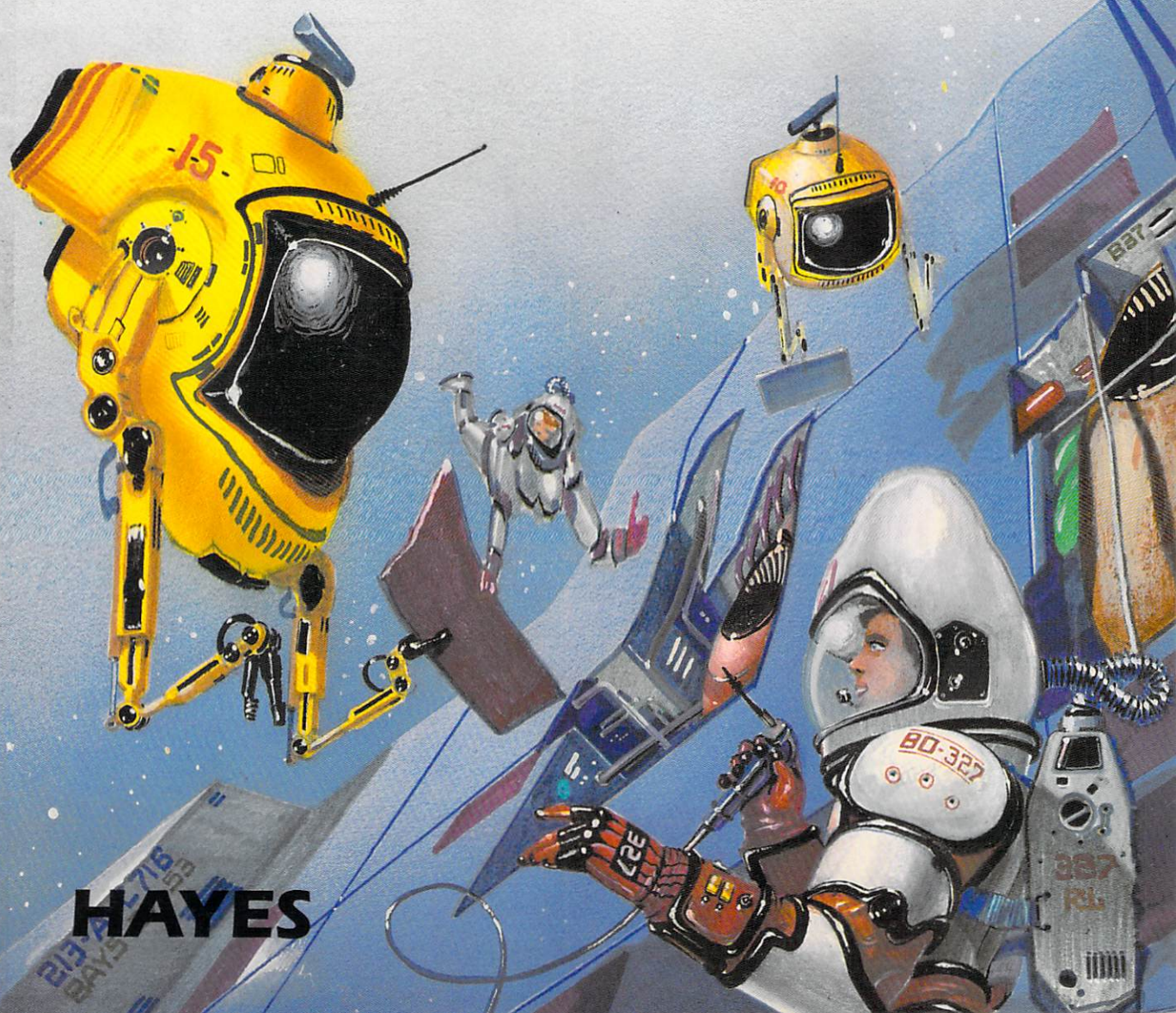


**GETTING
AHEAD**

SPACE ADVENTURES

for the VIC 20



SPACE ADVENTURES

for the VIC 20

Programs Compiled and Edited by Peter Lear

Editor:	Carol Batchelor, B.A., M.A.
Design:	Pat Strachan
Illustration:	Richard Livingston

NOTE

The publisher and authors have made every effort to assure that the computer programs are accurate and complete. However, this publication is prepared for general readership, and neither the publisher nor the authors have any knowledge about or ability to control any third party's use of the programs and programming information. There is no warranty or representation by the publisher or the authors that the programs or programming information in this book will enable the reader or user to achieve any particular result.

Copyright © 1984 by Hayes Publishing Ltd. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without permission from the publisher.

© 1984 Hayes Publishing Ltd., Burlington, Ontario

ISBN 0-88625-058-7

VIC 20 is a registered trademark of Commodore Computer Systems.

CONTENTS

Introduction	4
MOONSHUTTLE	6
METEOR SHOWER	12
About DATA Statements	17
PROTECTOR	18
ALIEN ATTACK	24
About READ Statements	35
RED ALERT	36
INVASION	42
How A Game Is Made	48

INTRODUCTION

This book contains listings for games written in BASIC for the VIC 20 computer made by Commodore Business Machines.

Some of the games require memory expansion. The type of expansion required is noted before each listing.

TYPING IN THE PROGRAMS

The program listings are set up like the listings in your VIC 20 owner's manual. To make the keying of the programs easier, a

standard has been used. Whenever you encounter square brackets ([]) in a program listing, refer to the following guide.

To do this:	Type this:	
[SC]	SHIFT	AND CLR HOME
[CU]	SHIFT	AND CRSR
[CL]	SHIFT	AND CRSR
[IN]	SHIFT	AND INST/DEL
[RV]	CTRL	AND) 9
[RD]	CTRL	AND # 3
[RO]	CTRL	AND 0
[f]	CTRL	AND ←
[BL]	CTRL	AND ' 7

To do this:	Type this:	
[YL]	CTRL	AND (8
[BK]	CTRL	AND ! 1
[PU]	CTRL	AND % 5
[CY]	CTRL	AND \$ 4
[GR]	CTRL	AND & 6
[WH]	CTRL	AND " 2
[CR]	CRSR	
[CD]	CRSR	
[HM]	CLR HOME	

When typing in these programs you can save some time by using the Commodore abbreviations for some of the BASIC language words. You know from reading your owner's manual that the question mark (?) is a short way of typing PRINT. Most

other BASIC words can be shortened by typing in the first letter normally, followed by the second letter together with the shift key. This will cause a graphic character to appear instead of the second letter. The VIC will interpret this as an entire BASIC word. Here are some examples:

To Get	You Type	As It Appears On The Screen
RUN	R shifted U	R □
LIST	L shifted I	L □
GOTO	G shifted O	G □

There are exceptions to this rule. The words RETURN, RESTORE, GOSUB, CLOSE, CLR, LET, STEP, and STR\$ can be

abbreviated by typing the first two letters normally and shifting the third.

To Get	You Type	As It Appears On The Screen
GOSUB	GO shifted S	GO ♥
RETURN	RE shifted T	RE □

The words IF, ON, ST and TI cannot be shortened.

Accidents do happen, you could type in most of a program only to have a power failure and lose it all, or you could accidentally RUN a program with mistakes

in it that destroy the program. So SAVE copies of the games on tape or floppy disk as you enter them. Do so often. Once you have a final copy of the game, you can record over the other versions on tape or delete the disk versions.

DEBUGGING

After typing in a program and SAVEing it, the VIC may have trouble RUNning it. This sometimes happens when copying a program from paper to computer. Check that you have entered all the program lines correctly and completely.

A common problem is the famous SYNTAX ERROR. If you get one, LIST the program line on the screen. Compare it to the book listing. You will probably see one of these problems:

1. Spelling error, or
2. Punctuation error (brackets, commas, colons or semi-colons missing), or
3. The number zero (0) confused with the letter 'O' (or vice versa), or

4. The number one (1) confused with the letter 'T' (also vice versa).

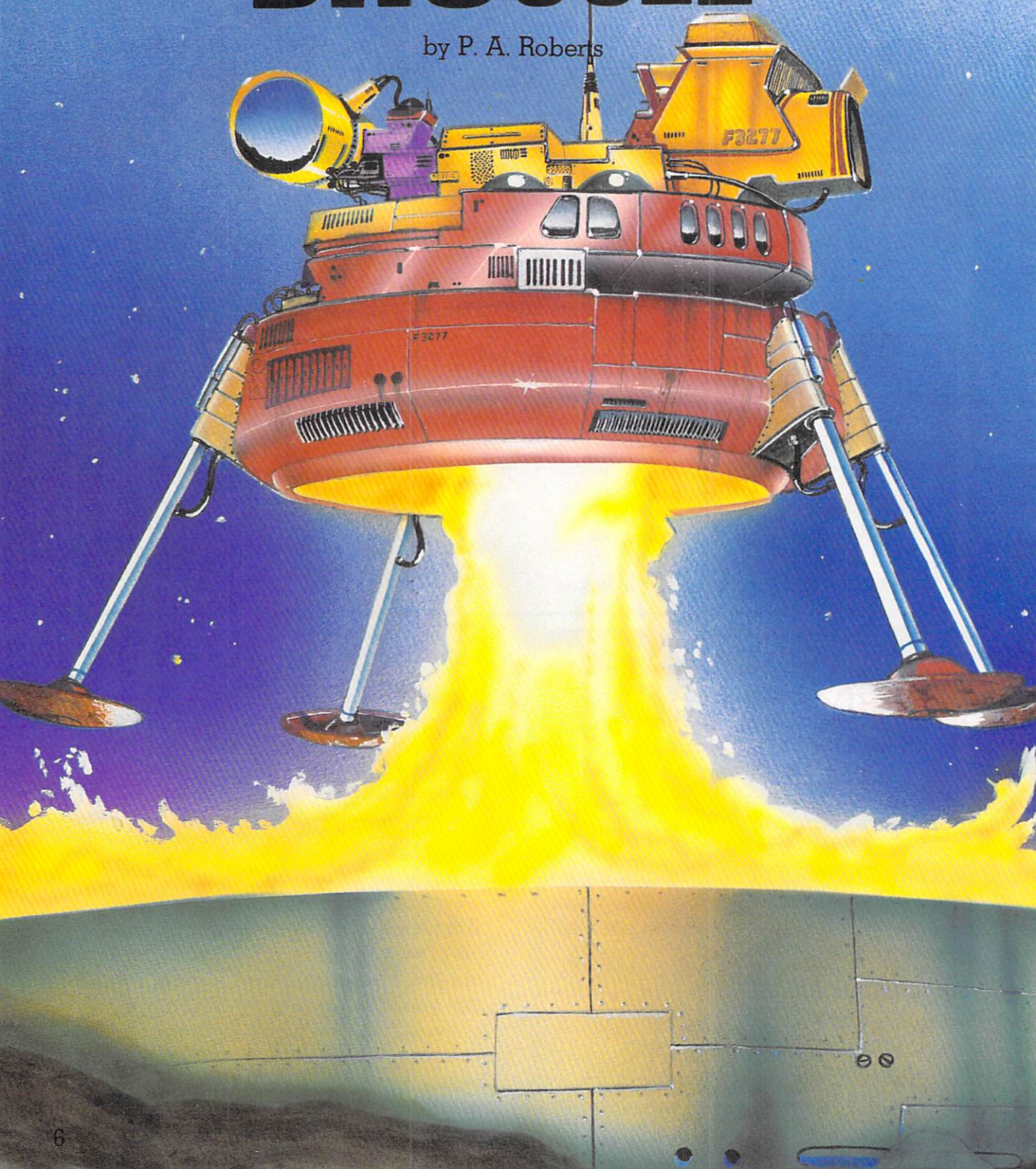
Fix the error and RUN the program again. It may take several attempts to get all the bugs out, but the work will be worth it. Remember to SAVE a final copy that has all the corrections made.

EXPERIMENT

If you have some programming knowledge, do not be afraid to try changing the games. After each program description there is a list of the program's variables and how they are used. And remember, there is nothing that you can enter into the VIC by program or through the keyboard that can do any permanent damage. If something goes wrong, you can turn the VIC off and on to begin again.

MOON SHUTTLE

by P. A. Roberts



Expansion required: None

Five...four...three...two...one...lift off!



ou have just left Phobos, one of the two moons of the planet Mars, and now you must land on Deimos, the other moon. But landing here isn't an easy task. You must navigate your moonshuttle through a deadly asteroid field. Using your four thrusters, U to go up, J to move right, H to move left and N to go down, you can maneuver through your descent and avoid the asteroids. To keep track of your fuel and energy consumption, read-outs of this information are given periodically.

Special Note: This game was designed to be saved on a data cassette and consists of two programs. The first one sets up the special graphics and gives instructions. The second is the game. Both must be entered separately to have the full effect of the graphics. When RUNning the programs, run the setup program first. It will automatically start the second program.

If you are using a disk drive instead of a data cassette, change Line 500 to read:

```
500 PRINT " [SC] [CD] [CD] [CD]
LOAD";CHR$(34);
"MOONSHUTTLE";CHR$(34);
CHR$(44);"8 [CD] [CD] [CD]
[CD] [CD] [CD] [CD] [CD] [CD]
[CD] [CR] [CR] [CR] [CR] [CR]
[CR] [CR] [CR] RUN [HM]"
```

The two programs will automatically run one after the other.

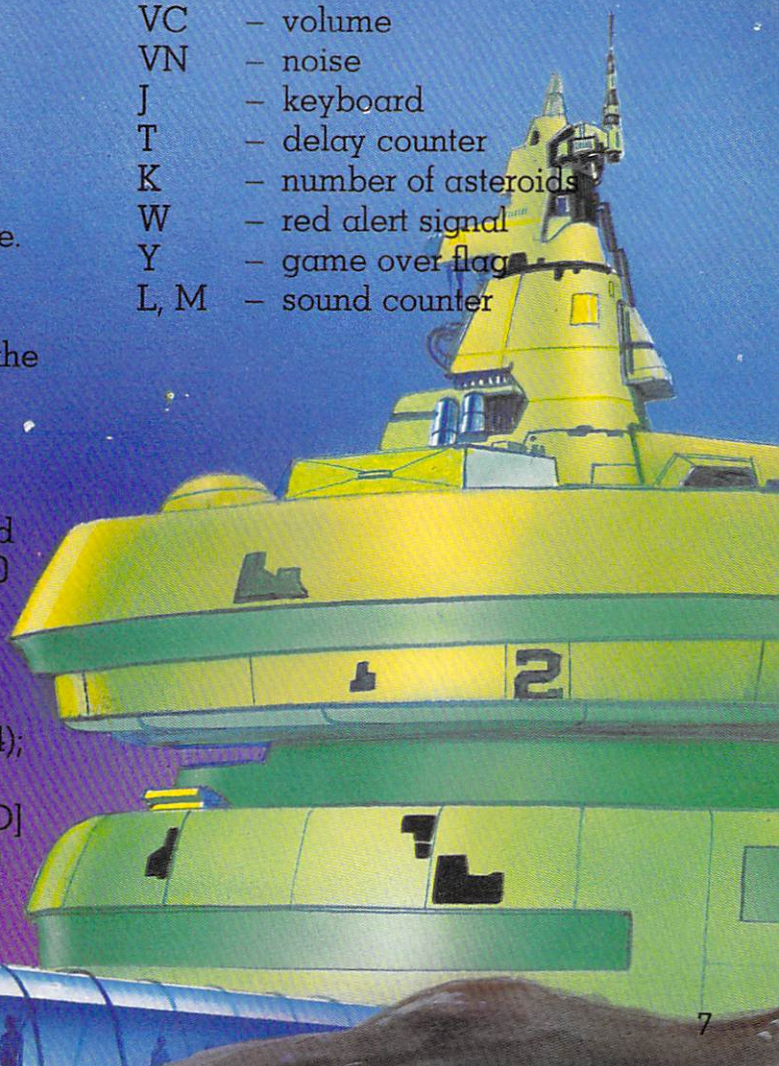
VARIABLES USED:

SETUP PROGRAM

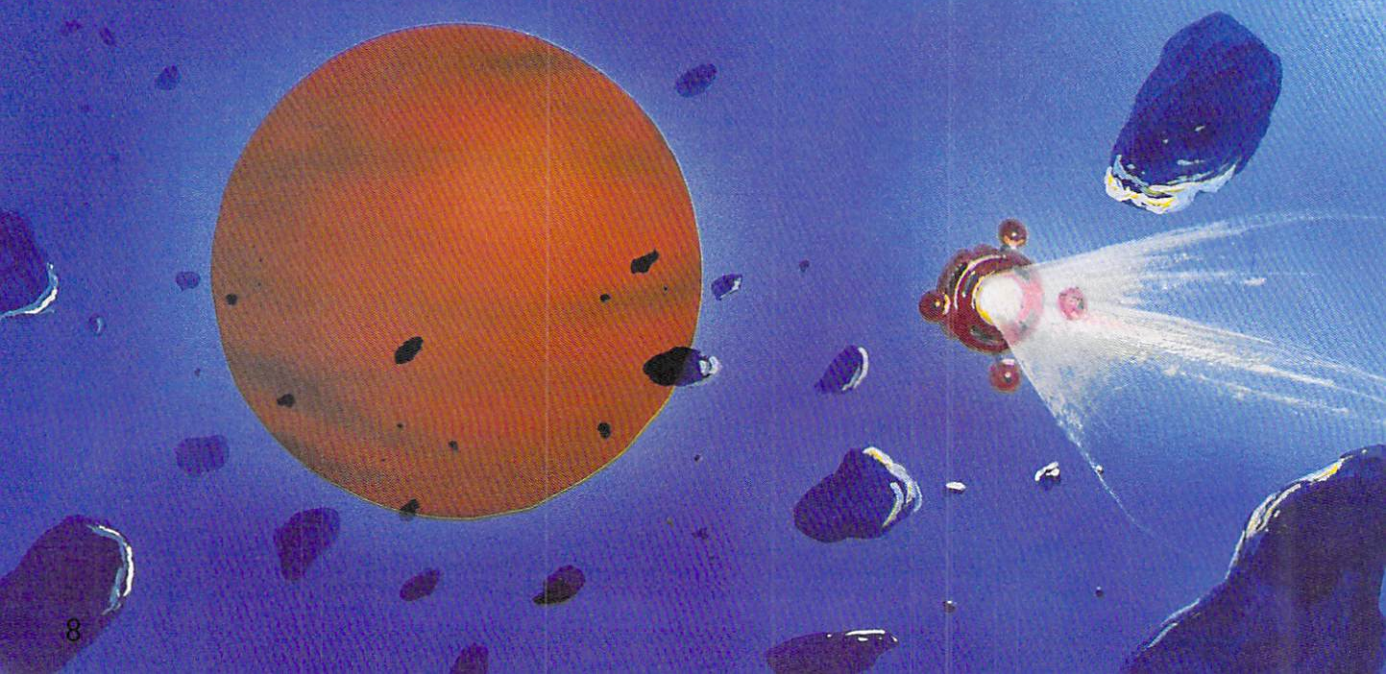
DT - start of character memory
CC - top of memory
I - counter
RE\$ - keyboard response

MAIN PROGRAM:

D, DI - difficulty factor
A, B - moonshuttle
Z - E (energy) meter
H - height
V - fuel
U, Q - color screen
P, U - moon base
VC - volume
VN - noise
J - keyboard
T - delay counter
K - number of asteroids
W - red alert signal
Y - game over flag
L, M - sound counter




```
100 PRINT"[SC][CD]WAIT ONE MOMENT..."
110 DT=32768:POKE52,28:POKE51,0:POKE56,28:POKE55,0
120 CC=256*PEEK(52)+PEEK(51)
130 FORI=0TO511:POKECC+I,PEEK(DT+I):NEXT
140 FORI=7168TO7207:READJ:POKEI,J:NEXTI
150 DATA 24,60,126,255,126,90,90,231
160 DATA 255,255,255,255,255,255,255,255
170 DATA 90,165,90,165,90,60,24,24
180 DATA 60,126,189,90,165,90,165,66
190 DATA 10,64,17,129,62,221,231,182
200 PRINT"[SC][CD][CD]"
210 PRINTTAB(6);"MOONSHUTTLE"
215 PRINTTAB(11);"BY"
220 PRINTTAB(6);"P A ROBERTS[CD]"
230 PRINT"YOUR MISSION IS TO LAND YOUR MOONSHUTTLE"
240 PRINT"SAFELY ON A BLUE LAND-ING PAD. USE THE KEYS:[CD]"
250 PRINTTAB(10);"UP"
260 PRINTTAB(10);"U"
270 PRINTTAB(4);"LEFT H J RIGHT"
280 PRINTTAB(10);"N"
290 PRINTTAB(8);"DOWN"
300 PRINT"[CD]PRESS A KEY TO LOAD"
310 POKE198,0
320 GETRES:IFRES=""THEN320
500 PRINT"[SC][HM][CD][CD][CD][CD]LOAD[CD][CD][CD][CD][CD]"
   [CD][CD][CD][CD][CD]RUN[HM]"
510 FORI=631TO640:POKEI,13:NEXT:POKE198,10
```




```

100 PRINT"[SC][CD][CD]DIFFICULTY (0-5)":INPUTD
110 DI=(D*20+10):POKE36869,255:GOTO280
130 IFPEEK(A)<>230ANDPEEK(A)<>32ANDPEEK(A)<>232THENW=2
140 IFPEEK(A)=46THENPRINT"[HM][PU][RV]SHIP HIT BADLY"
150 RETURN
160 IFA=8124ORA=8123THENY=2:E=E+10
170 IFA=8182THENY=2:E=E+60
180 IFA=8130THENY=2:E=E+30
190 IFA=8134THENY=2:E=E+20
200 RETURN
210 Z=Z-1:IFZ<=0THENPRINT"[HM][PU]OUT OF FUEL!"
230 RETURN
240 IFA<7856ANDV<>2THENPRINT"[HM][RV][CY]H=";H;"E=";Z
250 IFA>=8054ANDV<>2THENPRINT"[HM][RV][CY]H=";H/10;" [PU]
[RV]E-METER FAULTY!"
260 IFA>=7856ANDV<>2THENPRINT"[HM][RV][CY]H=";(H/10);
"E/10=";(Z/10)
270 RETURN
280 POKE36879,8:Q=38840:P=8120:FORW=38840TO38905:POKEW,1:
NEXTW
300 PRINT"[SC][RV]PRESS A KEY"
310 P=8120:POKEP,239:POKEP+7,239:POKEP+1,228:POKEP+8,228:
POKEP+12,228:POKEP+2,249:POKEP+6,249
320 POKEP+15,249:POKEP+5,226:POKEP+13,226
330 FORU=8142TO8185:POKEU,1:NEXTU
340 Q=38840:P=8120:FORW=38862TO38905:POKEW,1:NEXTW
350 POKEP+9,1:POKEP+11,1:FORW=P+16TOP+21:POKEW,1:NEXTW
360 POKEQ+3,6:POKEQ+4,6:POKEQ+10,6:POKEQ+14,6
370 POKEP+3,232:POKEP+4,232:POKEP+10,232:POKEP+14,232
380 POKEQ+62,6:POKEP+40,32:POKEP+18,32
390 POKEP+62,230
400 FORU=38836TO38418STEP-22:POKEU,3
410 POKE38880,3:POKE8160,0:IFPEEK(197)=64THEN410
415 PRINT"[HM]"
420 FORU=300TO0STEP-1:PRINT"[HM][CY]";INT(U/50):NEXTU
430 U=38880:P=8160:VN=36877:VC=36878:J=197:H=190:W=0:Y=0:
Z=23:V=0
440 POKEVC,15:POKEVN,220
450 POKEP,3:POKEU,2
460 P=P-22:U=U-22
470 POKEP,0:POKEU,3:FORT=1TO50:NEXTT
480 IFP=7720THEN510
490 POKEP,32:POKEP+22,32
500 GOTO450
510 POKEVN,0:POKEVC,0:U=U+22:P=P+22:POKEP,32
520 FORK=1TODI:M=INT((RND(1)*395)+1):POKE7724+M,46:
POKE38422+M,INT(RND(1)*6+1):NEXTK

```



```

550 A=7720:B=38440:W=0
560 IFV=2THENPRINT"[HM][PU][RV]OUT OF FUEL!"
570 GOSUB240
580 IFPEEK(J)=51THEN670
590 IFPEEK(J)=20THEN770
600 IFPEEK(J)=43THEN890
610 IFPEEK(J)=28THEN990
620 POKE37878,0
630 IFA>=7900THENPOKEA,32:A=A+1:B=B+1:GOSUB130:POKEA,0:
    POKEB,3
640 IFA<7900THENPOKEA,32:A=A-1:B=B-1:GOSUB130:POKEA,0:
    POKEB,3
650 IFW=2THEN1110
660 FORT=1TO40:NEXTT:GOTO560
670 POKEVC,15:POKEVN,220
680 GOSUB210:IFV=2THEN620
690 GOSUB840:A=A-22:B=B-22
700 IFA<7680THENA=7680:B=38400
710 GOSUB130:GOSUB160
720 POKEA,0:POKEB,3:H=H+10:GOSUB240
730 IFW=2THEN1110
740 IFY=2THEN1290
750 IFPEEK(J)<>51THENPOKEVC,0:POKEVN,0:GOTO580
760 GOTO680
770 POKEVC,15:POKEVN,220
780 GOSUB210:IFV=2THENGOTO620
790 GOSUB840:A=A+23:B=B+23
800 IFA<7680THENA=7680:B=38400
810 GOSUB130:GOSUB160
820 POKEA,0:POKEB,3:H=H-10:GOSUB240
830 GOTO850
840 POKEA+22,3:POKEB+22,2:FORT=1TO90:NEXTT:POKEA,32:
    POKEA+22,32:RETURN
850 IFW=2THEN1110
860 IFY=2THEN1290
870 IFPEEK(J)<>20THENPOKEVC,0:POKEVN,0:GOTO580
880 GOTO780
890 POKEVC,15:POKEVN,220
900 GOSUB210:IFV=2THEN620
910 GOSUB840:A=A+21:B=B+21
920 IFA<7680THENA=7680:B=38400
930 GOSUB130:GOSUB160
940 POKEA,0:POKEB,3:H=H-10:GOSUB240
950 IFW=2THEN1110
960 IFY=2THEN1290

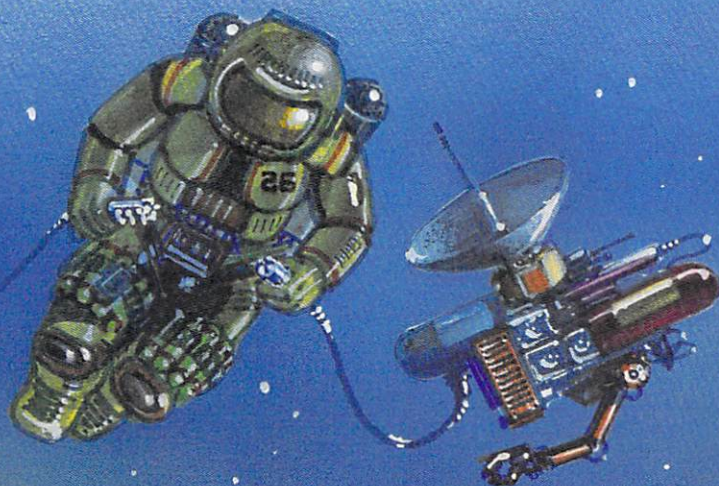
```




```

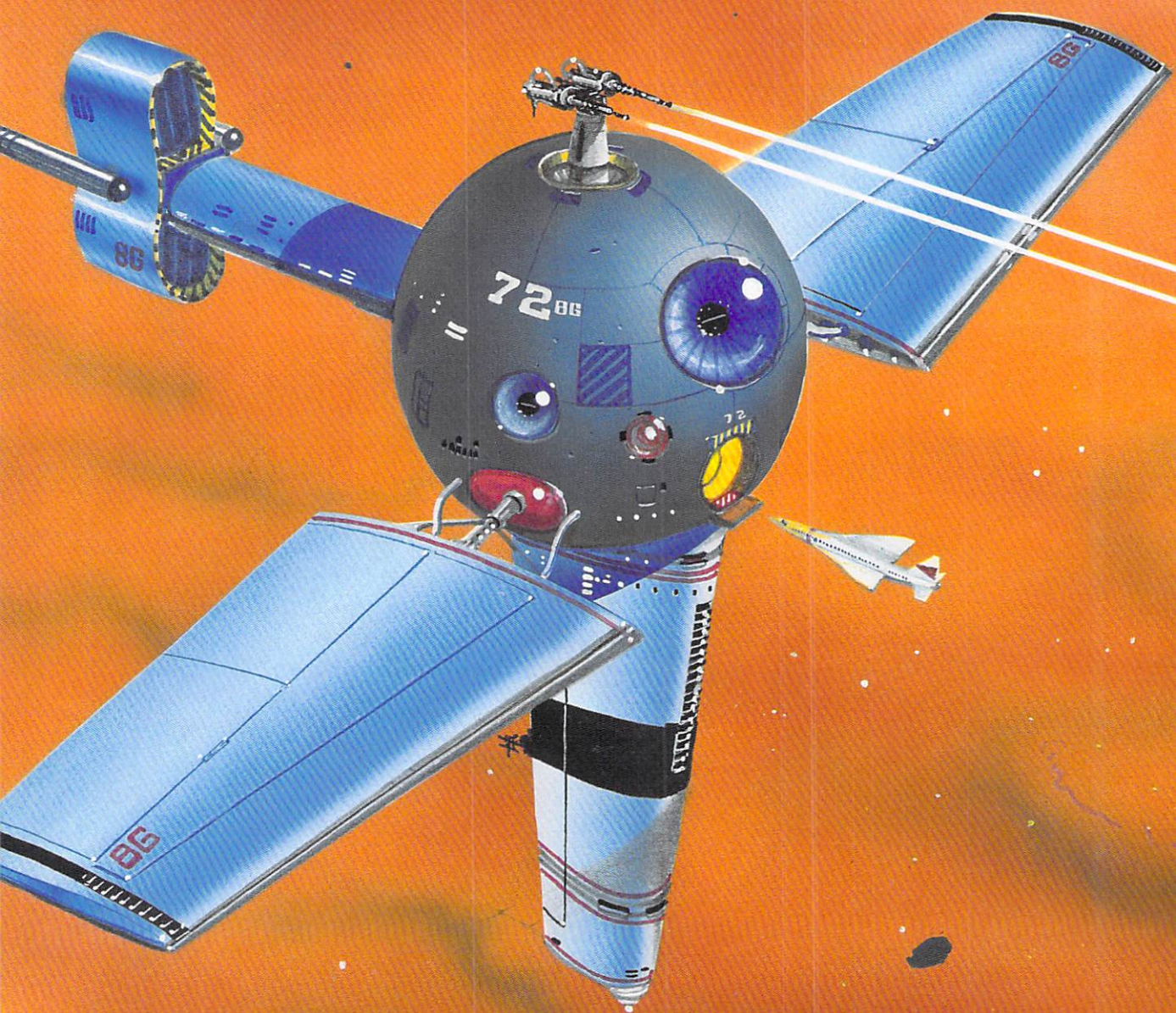
970 IF PEEK(J) <> 43 THEN POKEVC, 0: POKEVN, 0: GOTO 580
980 GOTO 900
990 POKEVC, 15: POKEVN, 220
1000 GOSUB 210: IFV=2 THEN 620
1010 A=A+22: B=B+22
1020 GOSUB 130: GOSUB 160
1030 A=A-22: B=B-22
1040 GOSUB 840: A=A+22: B=B+22
1050 IFA<7680 THEN A=7680: B=38400
1060 POKEA, 0: POKEB, 3: H=H-10: GOSUB 240
1070 IFW=2 THEN 1110
1080 IFY=2 THEN 1290
1090 IF PEEK(J) <> 28 THEN POKEVC, 0: POKEVN, 0: GOTO 580
1100 GOTO 1000
1110 POKEVN, 255: POKEVC, 15: PRINT "[HM][PU][RV] RED ALERT
      "
1120 U=INT((8142-A)/22)
1130 POKEVC, 15
1140 FOR L=UTO0 STEP -1: POKEA, 32: A=A+22: B=B+22: POKEA, 0:
      POKEB, INT(RND(1)*5+3)
1150 FOR T=1 TO 8: NEXT T: NEXT L
1160 POKEVN, 0
1170 FOR L=4 TO 0 STEP -1
1180 FORM=180 TO 235 STEP 2: POKE 36876, M
1190 POKEA, 0: POKEB, INT(RND(1)*5+3)
1200 NEXT M
1210 POKE 36876, 0
1220 NEXT L
1230 POKE 36876, 0
1240 POKEVN, 200: FOR L=15 TO 0 STEP -.05
1250 POKEA, 2: POKEB, INT(RND(1)*4+3)
1260 POKEVC, L: NEXT L: POKEVN, 0: POKEVC, 0
1270 POKEA, 4: POKEB, 3
1280 FOR T=1 TO 3000: NEXT T: PRINT "[SC]": GOTO 110
1290 PRINT "[HM][GR][RV] SAFE LANDING!      "
1300 POKEVC, 0: POKEVN, 0
1310 FOR T=1 TO 1000: NEXT T
1320 PRINT "[SC][RV][PU] SCORE="; E*DI; "    "
1330 FOR T=1 TO 1000: NEXT T: D=D+.5: GOTO 110

```



METEOR SHOWER

by Andrew Stanford and Peter Lear



Expansion required: 3K Memory Cartridge



n this shuttle mission you find your ship in a very heavy meteor shower.

There is nowhere to land to get out of it, so you'll just have to weather the storm as best you can. You can rotate your ship left and right using keys 1 and 2, and you can move forward by using key 9. Destroy meteors in your path with the zero key, your laser cannon. You may also jump into hyperspace momentarily with the space bar, but each hyperspace jump costs you 100 points! When your score reaches 300, watch out for flying saucers!

Note: To use a 3K Super Expander cartridge as a normal 3K cartridge, type the following line in before you load or enter the program:

POKE644,30:POKE643,0:SYS58232

To use an 8K expansion memory cartridge as a 3K, type the following line before you start the program:

POKE648,30:POKE642,32:SYS58648:SYS58232

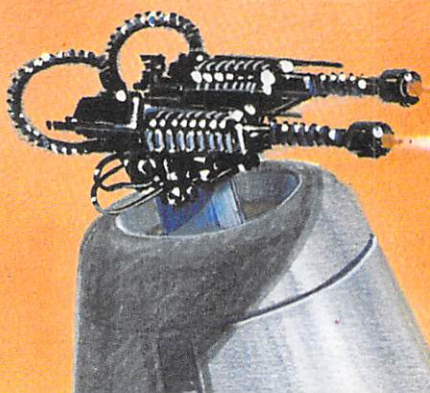
VARIABLES USED:

Z	- custom character counter
A	- general purpose counter
B	- data handler
SC	- score
F, F2, X	- ship position
D(1) -	
D(8), X2	- ship direction
M(1) -	
M(\$), R(1)	
- R(4)	- asteroid positions
M	- random number
T(1)	- T(4), U - UFO positions
J	- keyboard
RE\$	- keyboard response


```

100 DATA 191,2,191,2,191,2,207,8,223,8,0,4,219,2,217,2,213,
    2,231,8,223,8
110 DATA 0,4,219,2,217,2,213,2,231,8,223,8,219,2,217,2,219,
    2,213,12,0,4
120 DATA 16,16,16,16,16,16,16,16,0,0,0,255,0,0,0,0
130 DATA 64,0,1,16,0,64,1,130,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0
140 DATA 68,108,124,40,56,16,16,0,8,12,14,28,252,122,38,1
150 DATA 0,224,120,46,120,224,0,0,1,38,122,252,28,14,12,8
160 DATA 0,16,16,56,40,124,108,68,128,100,95,63,56,112,48,16
170 DATA 0,7,30,116,30,7,0,0,16,48,112,56,63,95,100,128
180 DATA 1,2,4,8,16,32,64,128,128,64,32,16,8,4,2,1
190 DATA 60,60,126,65,255,255,60,60,12,28,30,127,255,255,
    252,56
200 DATA 0,8,24,56,48,32,0,0,0,0,56,252,255,255,127
205 DATA 22,23,1,-21,-22,-23,-1,21
210 POKE52,28:POKE51,0:POKE56,28:POKE55,0:POKE36878,15
220 POKE36879,25:PRINT"[SC][CD][CR][CR][CR][CR][BL]*METEOR
    SHOWER*"
225 PRINT"[CD] BY ANDREW STANFORD & PETER LEAR"
230 PRINT"[CD][CD][CD][CD] <<CONTROLS>>"
240 PRINT"[CD] 1 AND 2 TURN SHIP":PRINT"[CD] 9 MOVES
    SHIP":PRINT"[CD] 0 FIRES A BEAM"
250 PRINT"[CD][CD] TO ENTER HYPERSPACE USE THE SPACE
    BAR"
260 FORZ=1TO22:READA:POKE36876,A:POKE36875,A
270 READB:FORX=1TOB*100:NEXT:POKE36876,0:POKE36875,0:NEXTZ
280 PRINT"[CD][CD]PRESS ANY KEY TO START"
290 GETA$:IFA$=""THEN290
300 FORA=232TO382:READB:POKE7168+A,B:NEXT
310 FORA=382TO470:POKE7168+A,PEEK(32768+A):NEXT
312 FORI=1TO8:READDX(I):NEXT

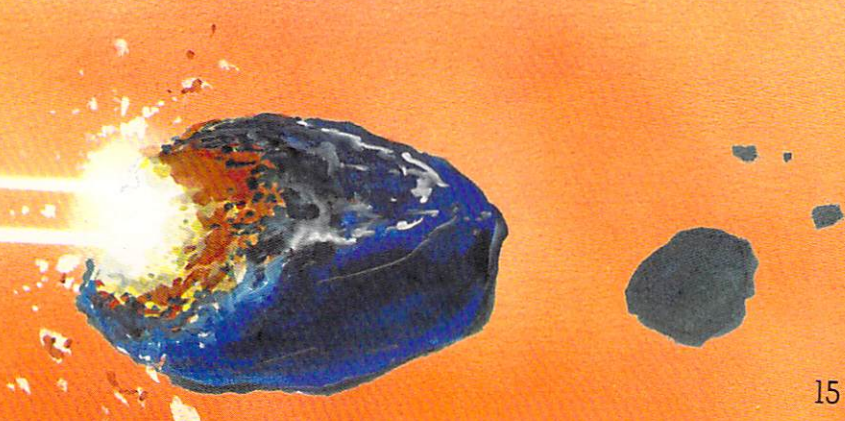
```




```

320 POKE36869,255
330 SC=0:X2=34:X=7901
350 POKE0,32
360 PRINT"[SC]":POKE36879,9
370 FORA=7658TO7679:POKEA,33:NEXT:FORA=8164TO8186:POKEA,33:
NEXT
380 FORA=7680TO8164STEP22:POKEA,33:NEXT
390 FORA=1TO4
395 FORAA=1TO2
400 IFPEEK(R(A))>33THEN540
410 IFA=1THENR(1)=INT(20*RND(1)+7703):T(1)=INT(3*RND(1)+45):
M(1)=INT(3*RND(1)+21)
420 IFA=1ANDSC>3000ANDINT(5*RND(1))=2THENT(1)=44:R(1)=7711:
M(1)=22
430 IFA=2THENR=INT(20*RND(1)):R(2)=7703+(R*22):T(2)=INT(3*
RND(1)+45):M=INT(6*RND(1))
440 IFA=2ANDM<1THENM(2)=23
450 IFA=2ANDM>1THENM(2)=-23
460 IFA=2ANDSC>3000ANDINT(5*RND(1))=2THENT(2)=44:R(2)=X+10:
M(2)=-1
470 IFA=3THENR=INT(20*RND(1)):R(3)=7723+(R*22):T(3)=INT(3*
RND(1)+45):M=INT(4*RND(1))
480 IFA=3THENM(3)=-21
490 IFA=3ANDM=2THENM(3)=21
500 IFA=3ANDM=1THENM(3)=-1
510 IFA=3ANDSC>3000ANDINT(5*RND(1))=2THENT(3)=44:R(3)=X-10:
M(3)=1

```



PROTECTOR

by L. Braine



Expansion required: Joystick



ou are on an intercept mission near the surface of Neptune. Only you and your ship stand between the space colony and the attacking aliens. They come in swarms of four at a time. Can even your valiant ship keep up this deadly pace? If you can ward off the aliens for two minutes, the colony will be saved. Good luck!

Special Note: This game was designed to be saved on a data cassette and consists of two programs. The first one sets up the special graphics and gives instructions. The second is the game. Both must be entered separately to have the full effect of the graphics. When RUNning the programs, run the setup program first. It will automatically start the second program.

If you are using a disk drive instead of a data cassette, change line 490 to read:

```
490 PRINT" [SC] [CD] [CD] [CD]
LOAD";CHR$(34);
"PROTECTOR";CHR$(34);
CHR$(44);"8 [CD] [CD] [CD]
[CD] [CD] [CD] [CD] [CD] [CD]
[CD] [CR] [CR] [CR] [CR] [CR]
[CR] [CR] [CR] RUN [HM]"
```

The two programs will automatically run one after the other.

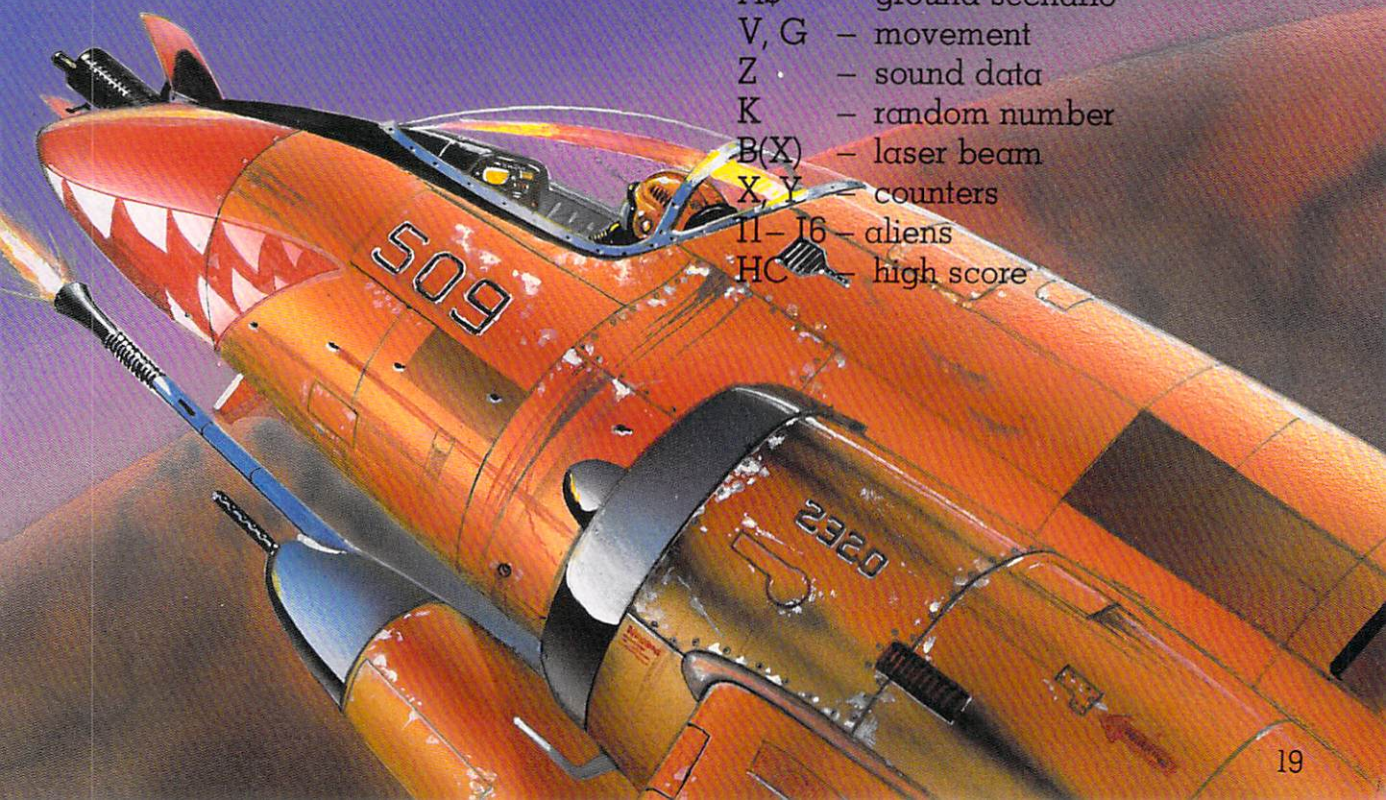
VARIABLES USED:

SETUP PROGRAM

I - counter
J - character data
RE\$ - keyboard response

MAIN PROGRAM

F, A,
PA, I - joystick
SC - score
SO - sound control
A\$ - ground scenario
V, G - movement
Z - sound data
K - random number
B(X) - laser beam
X, Y - counters
11-16 - aliens
HC - high score




```

100 POKE52,28:POKE56,28
105 PRINT"[SC][CD]PLEASE WAIT..."
110 DATA255,255,255,255,255,255,255,255
120 DATA0,0,0,0,0,0,0,255
130 DATA 0,0,0,0,0,0,255,255
140 DATA0,0,0,0,0,255,255,255
150 DATA0,0,0,0,255,255,255,255
160 DATA0,0,0,255,255,255,255,255
170 DATA0,0,255,255,255,255,255,255
180 DATA0,255,255,255,255,255,255,255
190 DATA126,90,255,189,189,36,66,36
200 DATA0,0,0,0,0,63,255,31,0
210 DATA0,12,126,255,255,254,254,0
220 DATA0,48,126,255,255,127,127,0
230 DATA0,0,0,0,252,255,248,0
240 DATA73,42,0,99,0,42,73,0
250 DATA1,3,7,15,31,63,127,255
260 DATA128,192,224,240,248,252,254,255
270 FORI=7168+32*8TO7168+32*8+7:POKEI,0:NEXTI
280 FORI=7168TO7295:READJ:POKEI,J:NEXTI
290 PRINT"[SC][CD]"
300 PRINTTAB(6);"PROTECTOR BY"
310 PRINTTAB(6);"L BRAINE"
320 PRINT"[CD]YOU HAVE TWO MINUTES TO DESTROY THE ALIENS"
330 PRINT"WITH YOUR VIPER FIGHT-ER(USE YOUR JOYSTICK)"
340 PRINT"PRESS ANY KEY TO LOAD"
350 POKE198,0
460 GETRE$:IFRE$=""THEN460
490 PRINT"[SC][CD][CD][CD][CD]LOAD[CD][CD][CD][CD][CD]
[CD][CD][CD][CD]RUN[HM]"
500 FORI=631TO640:POKEI,13:NEXT:POKE198,10

```




```

100 PRINT"[SC][CD][CD][CD][CD][CD][CD][CD][PU]          [RV]
    PROTECTOR"
110 PRINT"[CD]    PRESS FIRE BUTTON":PRINTTAB(8);"TO START"
120 CLR:GOTO240
130 POKE36869,255:POKE36878,15:POKE36879,29:PRINT"[SC]";:
    PA=37151:SO=36876:SC=0
140 PRINT"[HM][GR][WH][RV][BL]SCORE[GR][RO]@[RV][BL]TIME
    [GR][RO]@[RV][BL]HIScore[GR][RO]@@@@@@@@@@@@@@@@@@@@
    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
150 A$="[HM][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]
    [CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]"
160 S$="ABCDEFGH@OABCCBAN@GFEFGG@ONGFEDCBAABCDEFGH@OAAANOANGFED
    CDEFG@OAAAN@GFEDCBA"
170 PRINTLEFT$(A$,18);
180 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
    @@@@@@@@@@@@@@@@@@@@@@"
190 POKE8185,0:POKE38905,1
200 B$(1)="[BL]KL":B$(2)="[BL]IJ":F=1:G=10
210 FORA=1TO4:B(A)=7789+22*INT(RND(0)*9):NEXT
220 PRINTLEFT$(A$,20)"[RD]"LEFT$(S$,22)
230 RETURN
240 I=PEEK(37151):IF(IAND32)<>0THEN240
250 GOSUB130
260 TI$="0000000":PRINT"[HM][BL][CD][RV]"SPC(16)HC
270 PRINT"[HM][CD]"SPC(15)N$
280 FORX=1TO4
290 IFF=1THENGOSUB570
300 IFF=2THENGOSUB630
310 IFF=1THENGOSUB370
320 IFF=2THENGOSUB390
330 GOSUB410
340 PRINT"[HM][CD][RV][BL]"SPC(8)RIGHT$(TI$,3)
350 IFTI$="0000200"THEN960
360 NEXTX:GOTO280
370 T$=LEFT$(S$,1):S$=RIGHT$(S$,69)+T$
380 PRINTLEFT$(A$,20)"[RD]"LEFT$(S$,22);:RETURN
390 T$=RIGHT$(S$,1):T$=T$+LEFT$(S$,69):S$=T$
400 PRINTLEFT$(A$,20)"[RD]"LEFT$(S$,22);:RETURN

```




```

410 A=PEEK (PA)
420 IF (AAND32)=0 THEN 690: RETURN
430 POKESO, 0
440 IF (AAND16)=0 THEN 540
450 IF (AAND4)=0 THEN V=-1
460 IF (AAND8)=0 THEN V=1
470 IF G+V<5 OR G+V>17 THEN RETURN
480 IFF=1 THEN PRINT LEFT$(A$, G) "[CR].[CR]  "
490 IFF=2 THEN PRINT LEFT$(A$, G) SPC(18) "  "
500 G=G+V
510 IFF=1 THEN PRINT LEFT$(A$, G) "[CR][CR]" B$(F): RETURN
520 PRINT LEFT$(A$, G) SPC(18) B$(F)
530 RETURN
540 PRINT LEFT$(A$, G) "[CR][CR]  " SPC(14) "  "
550 IFF=1 THEN F=2: GOTO 520
560 F=1: GOTO 510
570 Z=21: K=RND(1)
580 IF K>.66 THEN Z=-1: GOTO 600
590 IF K>.33 THEN Z=-23
600 IF B(X)+Z<7767 OR B(X)+Z>8097 THEN Z=0: POKE B(X), 32: B(X)=7789
+22*INT(RND(0)*9): GOTO 620
610 POKE B(X), 32: B(X)=B(X)+Z
620 POKE B(X), 8: POKE B(X)+30720, 2: RETURN
630 Z=23: K=RND(1)
640 IF K>.66 THEN Z=1: GOTO 660
650 IF K>.23 THEN Z=-21
660 IF B(X)+Z<7767 OR B(X)+Z>8097 THEN Z=0: POKE B(X), 32: B(X)=7789
+22*INT(RND(0)*9): GOTO 680
670 POKE B(X), 32: B(X)=B(X)+Z
680 POKE B(X), 8: POKE B(X)+30720, 2: RETURN
690 TB=250: POKESO, 0: I3=0: IFF=2 THEN 830
700 FOR I1=7684+22*(G-1) TO 7696+22*(G-1): TB=TB-1
710 POKE I1, 198: POKE I1+30720, 6: POKESO, TB

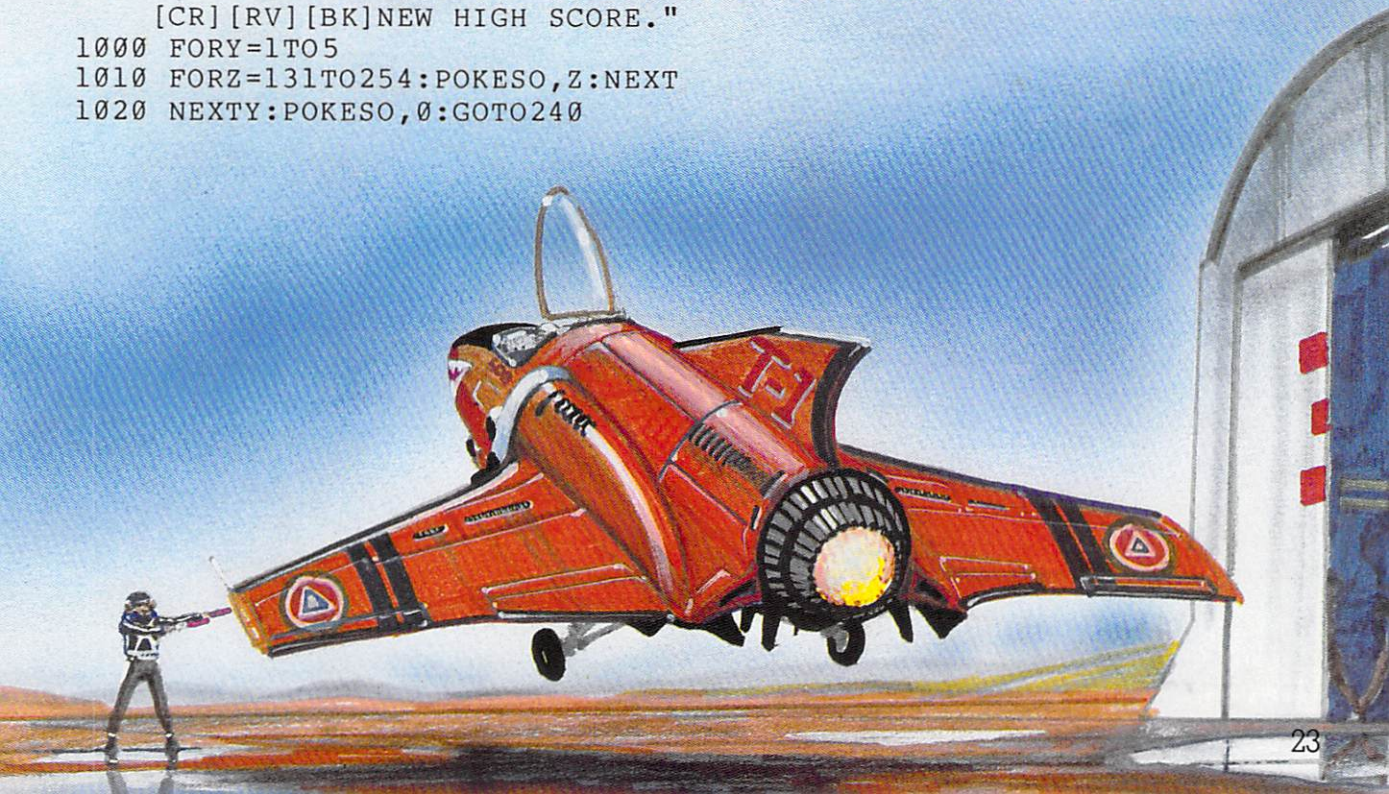
```




```

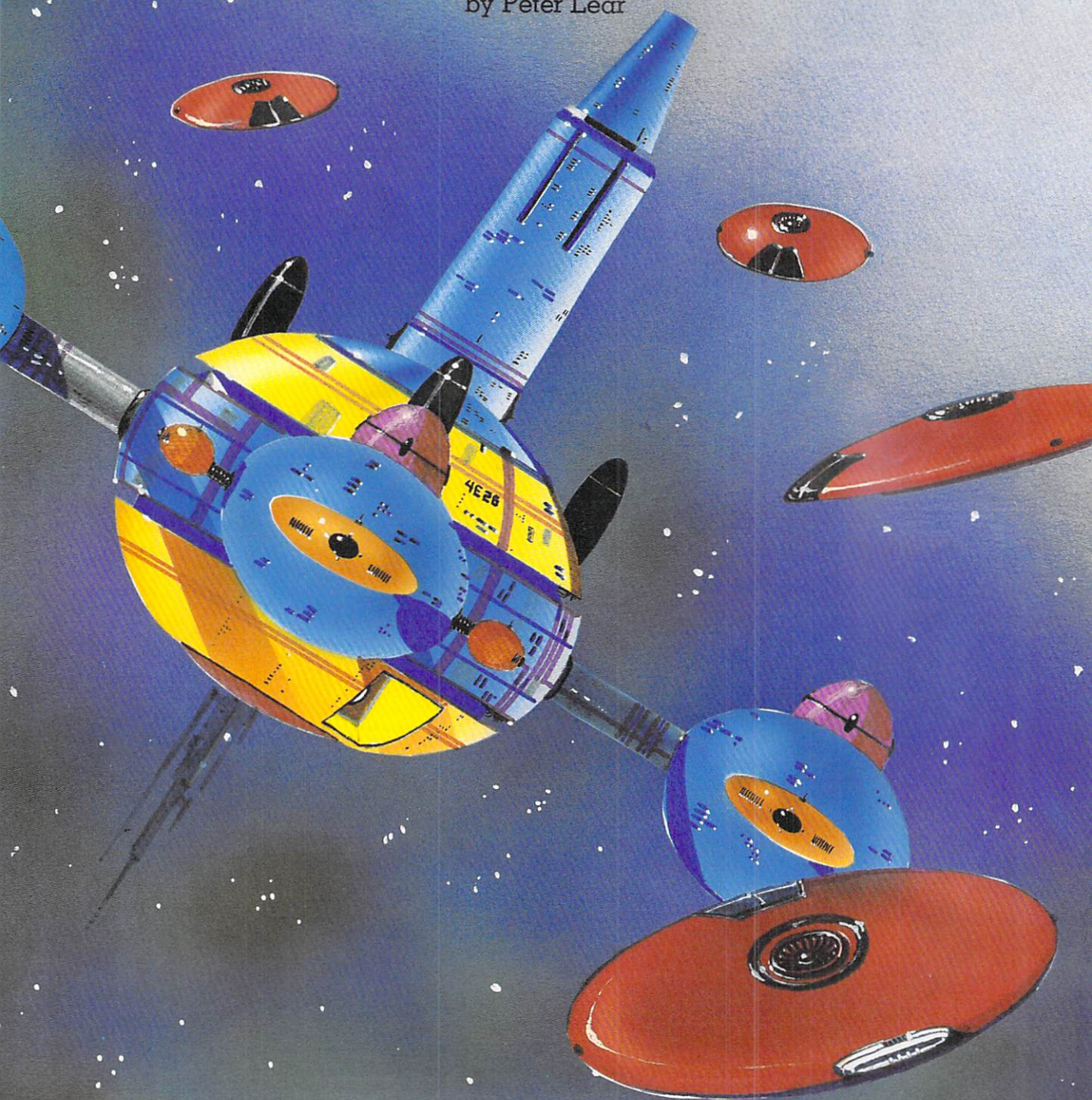
720 IFPEEK(I1+1)=8THENI2=I1-1:I1=7684+22*(G-1):I3=1:GOTO750
730 NEXT:POKESO,0
740 I2=7696+22*(G-1)
750 FORI4=7684+22*(G-1)TOI2:POKEI4,32:NEXT
760 IFI3=0THENRETURN
770 FORI5=1TO4:IFPEEK(B(I5)-1)=198THENI6=I5
780 NEXTI5
790 POKEB(I6),13:POKESO,250:FORZ=1TO99:NEXT
800 POKEB(I6),32:POKESO,0:POKEB(I6)-1,32
810 SC=SC+10:PRINT"[HM][CD][CR][BL][RV]"SC:IFSC>HCTHENPRINT
  "[HM][CD][RV]"SPC(16)SC:P=1
820 B(I6)=7789+22*INT(RND(0)*9):RETURN
830 FORI1=7697+22*(G-1)TO7685+22*(G-1)STEP-1:TB=TB-1
840 POKEI1,198:POKEI1+30720,6:POKESO,TB
850 IFPEEK(I1-1)=8THENI2=I1+1:I1=7685+22*(G-1):I3=1:GOTO880
860 NEXT
870 I2=7685+22*(G-1)
880 FORI4=7697+22*(G-1)TOI2STEP-1:POKEI4,32:NEXT
890 IFI3=0THENRETURN
900 FORI5=1TO4:IFPEEK(B(I5)+1)=198THENI6=I5
910 NEXTI5
920 POKEB(I6),13:POKESO,250:FORZ=1TO99:NEXT
930 SC=SC+10:PRINT"[HM][CD][CR][RV]"SC:IFSC>HCTHENPRINT
  "[HM][BL][CD][RV]"SPC(16)SC:P=1
940 POKEB(I6),32:POKESO,0:POKEB(I6)+1,32
950 B(I6)=7789+22*INT(RND(0)*9):RETURN
960 PRINT"[HM][CD][CD][CD][CD]";:FORB=1TO330:PRINT" ";:
  POKESO,INT(RND(0)*120)+131:NEXT:POKESO,0
970 IFSC<HCTHENPRINT"[RD][HM][RV][CD][CD][CD][CD][CD][CD]
  [CD][CR][CR][CR][CR][CR][CR][CR]GAME OVER":GOTO240
980 HC=SC
990 PRINT"[HM][CD][CD][CD][CD][CD][CD][CD][CD][CR][CR][CR]
  [CR][RV][BK]NEW HIGH SCORE."
1000 FORY=1TO5
1010 FORZ=131TO254:POKESO,Z:NEXT
1020 NEXTY:POKESO,0:GOTO240

```



ALIEN ATTACK

by Peter Leach



Expansion required: 8K (or greater)

Memory

Cartridge

Has Joystick

Option



ou await your doom inside the crippled space station. The aliens will be attacking in clusters.

There are eight different ways that they can get in. When they do come, the station's force shields will protect you from at least three blasts on each side. Then, if you're lucky you will figure out how to operate the malfunctioning laser cannon and fight back. There are several controls that could do it — your joystick and some keys. Try T, Y, U, G, J, B, N, and M. Maybe by destroying enough attackers you will be able to recharge your shield and save the space station and your life.

Note: Try pressing H, C and other keys after the game.

VARIABLES USED:

JK	- joystick option
I, II, J, K, KK, L	- counters
A\$	- general question response
JS(I,J), DD, PB,	
P,PA,S0,S1,	- joystick
S2,S3,FR	
DI	- direction of fire
S	- center of shields
M1	- laser origin
P1, P2	- score
C	- screen color
M(1) - M(8)	- enemy ships
D2	- move up
CF, R, A	- flags
D	- difficulty factor
E(1) - E(8)	- explosion data
K1 - K3	- temporary data store
A(1) - A(4),	
AA(1) - AA(8),	
B(1) - B(4)	- data handlers
PP	- player character




```

100 REM ALIEN ATTACK
110 REM BY PETER LEAR
140 PRINT"[SC]";CHR$(142)
150 POKE36876,0:POKE36874,0:POKE36879,8
160 READJK,D:IFJK>0THEN240
170 PRINT"[SC] [CR][CR][CR]ALIEN ATTACK"
180 PRINT"[CR][CR][CR][CR]BY PETER LEAR"
190 PRINT"[CD][CD][CD][CD][CR][CR][CR][CR]JOYSTICK(1)
      KEYBOARD(2)"
200 FORI=1TO10:GETA$:NEXTI
210 GETA$:IFVAL(A$)<1ORVAL(A$)>2THEN210
220 JK=VAL(A$)
230 IFJK=2THENGOSUB3370
240 GOTO2210
250 REM JOY CHECK
260 POKEDD,127:S3=((PEEK(PB)AND128)=0):POKEDD,255
270 P=PEEK(PA):S1=((PAND8)=0):S2=((PAND16)=0):S0=((PAND4)=0)
280 FR=((PAND32)=0):X=S2+S3:Y=S0+S1:DI=JS(X+1,Y+1):RETURN
290 REM KEYBOARD
300 IFPEEK(197)=50THENDI=1:RETURN
310 IFPEEK(197)=11THENDI=2:RETURN
320 IFPEEK(197)=51THENDI=3:RETURN
330 IFPEEK(197)=19THENDI=4:RETURN
340 IFPEEK(197)=36THENDI=9:RETURN
350 IFPEEK(197)=20THENDI=6:RETURN
360 IFPEEK(197)=35THENDI=7:RETURN
370 IFPEEK(197)=28THENDI=8:RETURN
380 DI=0:RETURN
390 REM -23
400 POKEM1-23,85:POKEM1+C-23,3
410 FORI=S+161TOS+22STEP-23
420 IFPEEK(I)>46THENM(1)=0:P1=P1+50:P2=P2+50:GOTO470
430 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
440 NEXTI
450 POKE36878,0
460 RETURN
470 POKES+22,32:POKES+23,32:POKES+1,32
480 POKE36878,0
490 RETURN
500 REM -22
510 POKEM1-22,114:POKEM1+C-22,3
520 FORI=S+120TOS+8STEP-22
530 IFPEEK(I)>46THENM(2)=0:P1=P1+75:P2=P2+75:GOTO580
540 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
550 NEXTI
560 POKE36878,0

```



```

570 RETURN
580 FORI=9TO11:POKES+I,32:NEXTI:POKES+32,32
590 POKE36878,0
600 RETURN
610 REM -21
620 POKEM1-21,73:POKEM1+C-21,3
630 FORI=S+189TOS+21STEP-21
640 IFPEEK(I)>46THENM(3)=0:P1=P1+50:P2=P2+50:GOTO690
650 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
660 NEXTI
670 POKE36878,0
680 RETURN
690 POKES+20,32:POKES+42,32:POKES+43,32
700 POKE36878,0
710 RETURN
720 REM -1
730 POKEM1-1,107:POKEM1+C-1,3
740 FORI=S+246TOS+242STEP-1
750 IFPEEK(I)>46THENM(8)=0:P1=P1+75:P2=P2+75:GOTO800
760 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
770 NEXTI
780 POKE36878,0
790 RETURN
800 FORI=220TO264STEP22:POKES+I,32:NEXTI:POKES+242,32
810 POKE36878,0
820 RETURN
830 POKEM1,90:POKEM1+C,1
840 RETURN
850 REM 1
860 POKEM1+1,115:POKEM1+C+1,3
870 FORI=S+258TOS+263STEP1
880 IFPEEK(I)>46THENM(4)=0:P1=P1+75:P2=P2+75:GOTO930
890 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
900 NEXTI
910 POKE36878,0
920 RETURN
930 FORI=241TO285STEP22:POKES+I,32:NEXTI:POKES+262,32
940 POKE36878,0
950 RETURN
960 REM 21
970 POKEM1+21,74:POKEM1+C+21,3
980 FORI=S+315TOS+500STEP21
990 IFPEEK(I)>46THENM(7)=0:P1=P1+50:P2=P2+50:GOTO1040
1000 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
1010 NEXTI

```



```

1020 POKE36878,0
1030 RETURN
1040 POKES+440,32:POKES+441,32:POKES+463,32
1050 POKE36878,0
1060 RETURN
1070 REM 22
1080 POKEM1+22,113:POKEM1+C+22,3
1090 FORI=S+384TOS+500STEP22
1100 IFPEEK(I)>46THENM(6)=0:P1=P1+75:P2=P2+75:GOTO1150
1110 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
1120 NEXTI
1130 POKE36878,0
1140 RETURN
1150 POKES+493,32:POKES+494,32:POKES+495,32
1160 POKE36878,0
1170 RETURN
1180 REM 23
1190 POKEM1+23,75:POKEM1+C+23,3
1200 FORI=S+321TOS+505STEP23
1210 IFPEEK(I)>46THENM(5)=0:P1=P1+50:P2=P2+50:GOTO1260
1220 POKEI,42:POKEI+C,2:POKEI,32:POKE36878,15
1230 NEXTI
1240 POKE36878,0
1250 RETURN
1260 POKES+481,32:POKES+459,32:POKES+460,32
1270 POKE36878,0
1280 RETURN
1290 REM TOP LEFT
1300 POKE36877,190:POKE36878,15
1310 POKES+22,105:POKES+C+22,5:POKES+23,75:POKES+C+23,5:POKE
    S+1,105:POKES+C+1,5
1320 IFM(1)=1THENM(1)=2:POKE36877,252:POKE36878,0:RETURN
1330 FORI=S+91TOS+264STEP23
1340 IFI=S+252THEN2710
1350 FORII=-1TO1
1360 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+252
1370 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
1380 NEXTII,I
1390 POKE36877,252:POKE36878,0
1400 RETURN
1410 REM TOP MIDDLE
1420 FORI=9TO11:POKES+I,104:POKES+I+C,5:NEXTI:POKES+32,120:
    POKES+C+32,5
1430 POKE36877,190:POKE36878,15
1440 IFM(2)=1THENM(2)=2:POKE36877,252:POKE36878,0:RETURN
1450 FORI=S+54TOS+264STEP22

```



```

1460 IFI=S+252THEN2710
1470 IFPEEK(I)<>32THENPOKEI,32:I=S+252
1480 POKEI,81:POKEI+C,2:POKEI,32
1490 NEXTI
1500 POKE36877,252:POKE36878,0
1510 RETURN
1520 REM TOP RIGHT
1530 POKE36877,190:POKE36878,15
1540 POKES+20,95:POKES+C+20,5:POKES+42,74:POKES+C+42,5:POKE
S+43,95:POKES+C+43,5
1550 IFM(3)=1THENM(3)=2:POKE36877,252:POKE36878,0:RETURN
1560 FORI=S+63TOS+252STEP21
1570 IFI=S+252THEN2710
1580 FORII=-1TO1
1590 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+252
1600 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
1610 NEXTII,I
1620 POKE36877,252:POKE36878,0
1630 RETURN
1640 REM MIDDLE RIGHT
1650 FORI=241TO285STEP22:POKES+I,92:POKES+I+C,5:NEXTI:POKE
S+262,118:POKES+C+262,5
1660 POKE36877,190:POKE36878,15
1670 IFM(4)=1THENM(4)=2:POKE36877,252:POKE36878,0:RETURN
1680 FORI=S+260TOS+252STEP-1
1690 IFI=S+252THEN2710
1700 IFPEEK(I)<>32THENPOKEI,32:I=S+252
1710 POKEI,81:POKEI+C,2:POKEI,32
1720 NEXTI
1730 POKE36877,252:POKE36878,0
1740 RETURN
1750 REM BOTTOM RIGHT
1760 POKE36877,190:POKE36878,15
1770 POKES+481,233:POKES+C+481,5:POKES+459,85:POKES+C+459,5:
POKES+460,233:POKES+C+460,5
1780 IFM(5)=1THENM(5)=2:POKE36877,252:POKE36878,0:RETURN
1790 FORI=S+436TOS+252STEP-23
1800 IFI=S+252THEN2710
1810 FORII=-1TO1
1820 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+252
1830 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
1840 NEXTII,I
1850 POKE36877,252:POKE36878,0
1860 RETURN
1870 REM BOTTOM MIDDLE

```



```

1880 POKES+493,104:POKES+493+C,5:POKES+494,102:POKES+C+494,5:
    POKES+495,104:POKES+C+495,5
1890 POKE36877,190:POKE36878,15
1900 IFM(6)=1THENM(6)=2:POKE36877,252:POKE36878,0:RETURN
1910 FORI=S+472TOS+252STEP-22
1920 IFI=S+252THEN2710
1930 IFPEEK(I)<>32THENPOKEI,32:I=S+252
1940 POKEI,81:POKEI+C,2:POKEI,32
1950 NEXTI
1960 POKE36877,252:POKE36878,0
1970 RETURN
1980 REM BOTTOM LEFT
1990 POKE36877,190:POKE36878,15
2000 POKES+440,223:POKES+C+440,5:POKES+441,73:POKES+C+441,5:
    POKES+463,223:POKES+C+463,5
2010 IFM(7)=1THENM(7)=2:POKE36877,252:POKE36878,0:RETURN
2020 FORI=S+420TOS+252STEP-21
2030 IFI=S+252THEN2710
2040 FORII=-1TO1
2050 IFPEEK(I+II)<>32THENPOKEI+II,32:I=S+252
2060 POKEI+II,81:POKEI+II+C,2:POKEI+II,32
2070 NEXTII,I
2080 POKE36877,252:POKE36878,0
2090 RETURN
2100 REM MIDDLE LEFT
2110 FORI=220TO264STEP22:POKES+I,92:POKES+I+C,5:NEXTI:POKE
    S+242,102:POKES+C+242,5
2120 POKE36877,190:POKE36878,15
2130 IFM(8)=1THENM(8)=2:POKE36877,252:POKE36878,0:RETURN
2140 FORI=S+246TOS+252STEP1
2150 IFI=S+252THEN2710
2160 IFPEEK((I)<>32THENPOKEI,32:I=S+252
2170 POKEI,81:POKEI+C,2:POKEI,32
2180 NEXTI
2190 POKE36877,252:POKE36878,0
2200 RETURN
2210 REM SET UP

```



```

2220 S=4*(PEEK(36866)AND128)+64*(PEEK(36869)AND120):C=37888
      -S+4*(PEEK(36866)AND128)
2230 DIMJS(2,2):POKE37139,0:DD=37154:PA=37137:PB=37152:A=2
2240 FORI=0TO2:FORJ=0TO2:READJS(J,I):NEXTJ,I
2250 FORI=1TO9:READHS(I):NEXTI
2260 FORI=1TO9:READD(I):NEXTI
2270 M1=S+252:R=7:PP=113:D2=22:CF=1
2280 POKE36877,252:POKE36875,152
2290 IFD>0THEN2340
2300 PRINT"[SC][CD][CD][CD][CD][CD][CD][CD][CD][CD]SHOTS FACTOR
      (1TO9)"
2310 GETA$
2320 IFVAL(A$)<1THEN2310
2330 D=VAL(A$)
2340 PRINT"[SC]"
2350 FORI=251TO253:POKEI+S,32:NEXTI
2360 FORI=272TO276:POKEI+S,32:NEXTI
2370 FORI=142TO186STEP22:POKES+I,100:POKES+C+I,1:NEXTI
2380 FORI=205TO207:POKES+I,78:POKES+C+I,1:NEXTI
2390 FORI=209TO211:POKES+I,77:POKES+C+I,1:NEXTI
2400 FORI=226TO228:POKES+I,78:POKES+C+I,1:NEXTI
2410 FORI=232TO234:POKES+I,77:POKES+C+I,1:NEXTI
2420 FORI=247TO249:POKES+I,103:POKES+C+I,1:NEXTI
2430 FORI=255TO257:POKES+I,101:POKES+C+I,1:NEXTI
2440 FORI=270TO272:POKES+I,77:POKES+C+I,1:NEXTI
2450 FORI=276TO278:POKES+I,78:POKES+C+I,1:NEXTI
2460 FORI=293TO295:POKES+I,77:POKES+C+I,1:NEXTI
2470 FORI=297TO299:POKES+I,78:POKES+C+I,1:NEXTI
2480 FORI=318TO362STEP22:POKES+I,99:POKES+C+I,1:NEXTI
2490 POKES+185,78:POKES+C+185,1:POKES+187,77:POKES+C+187,1
2500 POKES+317,77:POKES+C+317,1:POKES+319,78:POKES+C+319,1
2510 POKES+163,78:POKES+C+163,1:POKES+165,77:POKES+C+165,1
2520 POKES+184,78:POKES+C+184,1:POKES+188,77:POKES+C+188,1
2530 POKES+316,77:POKES+C+316,1:POKES+320,78:POKES+C+320,1
2540 POKES+339,77:POKES+C+339,1:POKES+341,78:POKES+C+341,1
2550 IFP2>0THENRETURN
2560 PRINT"[HM][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR]"

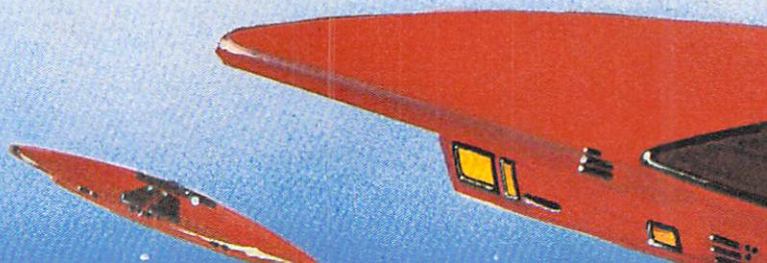
```



```

[CR][CR][CU]"HS(D)
2570 TI$="000000"
2580 REM MAIN PLAY
2590 PRINT"[HM][CR][CR]"P2
2600 ONJGOSUB260,300
2610 ONDIGOSUB400,510,620,730,830,860,970,1080,1190
2620 POKEM1+D(DI),32
2630 IFP1=>1200THENP1=P1-1200:GOSUB2370
2640 A=A*2:IFA<D*2THEN2590
2650 FORJ=1TOINT(P2/1000)STEP2
2660 M=INT(RND(TI)*(12+5*INT(P2/1000))):IFM>2+5*INT(P2/1000)
    THENM((12+5*INT(P2/1000))-M)=1
2670 FORM2=1TO8
2680 IFM(M2)>0THENONM2GOSUB1300,1420,1530,1650,1760,1880,
    1990,2110
2690 NEXTM2,J
2700 A=2:GOTO2590
2710 REM EXPLOSION
2720 A$=MID$(TI$,4,1)+":"+RIGHT$(TI$,2)
2730 FORII=1TO8:E(II)=S+274:READA(II):NEXTII
2740 FORII=1TO8:E(II)=S+274:READB(II):NEXTII
2750 POKE36875,0:POKE36877,0:POKE36878,15
2760 FORJ=1TO5
2770 FORII=1TO8:E(II)=A(II)+E(II):NEXTII
2780 FORII=1TO8:POKEE(II),32:NEXTII
2790 FORII=1TO8:POKEE(II),B(II):NEXTII
2800 POKE36877,225
2810 NEXTJ
2820 FORJ=1TO5

```




```

2830 FORII=1TO8:POKEE(II),32:NEXTII
2840 FORII=1TO8:E(II)=E(II)-A(II):NEXTII
2850 NEXTJ
2860 FORJ=142TO247STEP21:POKES+J,32:POKES+J+115,32:NEXTJ
2870 FORJ=165TO234STEP23:POKES+J,32:POKES+J+105,32:NEXTJ
2880 FORJ=164TO248STEP21:POKES+J,32:POKES+J+92,32:NEXTJ
2890 FORJ=187TO233STEP23:POKES+J,32:POKES+J+84,32:NEXTJ
2900 FORJ=186TO255STEP23:POKES+J,32:POKES+J+63,32:NEXTJ
2910 FORJ=207TO228STEP21:POKES+J,32:POKES+J+69,32:NEXTJ
2920 E(1)=S+251:E(2)=S+253:E(3)=S+297:E(4)=S+295
2930 FORII=1TO4:READA(II),B(II):NEXTII
2940 FORII=1TO8:READAA(II):POKE36879,AA(II):FORG=1TO25:
NEXTG,II
2950 POKE36877,220
2960 FORL=15TO0STEP-3
2970 FORII=1TO4:E(II)=E(II)-A(II):POKEE(II),B(II):NEXTII
2980 POKE36878,L
2990 REM FORM=1TO300
3000 FORII=1TO4:POKEE(II),32:NEXTII
3010 NEXTL
3020 POKE36877,0
3030 IFP2>HS(D)THENHS(D)=P2
3040 I=0:II=0:J=0:KK=0:M2=0
3050 PRINT"[HM][CR][CR][CR]"P2"[CD][CD][CD]"A$;:P2=0:P1=0
3060 PRINT"[HM][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR][CR]"HS(D)
3070 FORJ=1TO10:GETA$:NEXTJ
3080 GETA$
3090 POKE37139,128

```



```

3100 IFA$="H"THENGOSUB3250
3110 IFA$="C"THEND=0:GOTO3150
3120 IFA$<>" THEN3150
3130 POKE37139,0:IF (PEEK (PA)AND32)=0THEN3150
3140 GOTO3080
3150 PRINT"[SC][CD][CD][CD][CD][CD]"
3160 K=2:K1=89:K3=56
3170 PRINT"3410 DATA";JK;"[CL],";D
3180 PRINT"3430 DATA";
3190 FORK=1TO9:PRINTHS(K)"[CL],";:NEXTK:PRINTCHR$(20)
3200 PRINT"RUN"
3210 PRINT"[HM]"
3220 POKE198,10
3230 FORI=631TO640:POKEI,13:NEXTI
3240 STOP
3250 PRINT"[SC][CD]"
3260 PRINT"  LEVEL  "; " HI SCORE":PRINT
3270 FORK=1TO9
3280 PRINT"    "K,HS(K)
3290 PRINT
3300 NEXTK
3310 FORJ=1TO10:GETA$:NEXTJ
3320 GETA$
3330 POKE37139,128
3340 IFA$<>" THENRETURN
3350 POKE37139,0:IF (PEEK (PA)AND32)=0THENRETURN
3360 GOTO3320
3370 PRINT"[SC][CD][CD][CD]    USE THE KEYS"
3380 PRINT"[CD]      T Y U"
3390 PRINT"[CD]      G   J"
3400 PRINT"[CD]      B N M"
3401 POKE198,0
3403 GETDE$:IFDE$="" THEN3403
3404 RETURN
3410 DATA 0,0
3420 DATA 1,2,3,4,5,6,7,8,9
3430 DATA 0,0,0,0,0,0,0,0,0
3440 DATA -23,-22,-21,-1,0,1,21,22,23
3450 DATA -23,-22,-21,1,23,22,21,-1
3460 DATA 77,66,78,64,77,66,78,64
3470 DATA 21,77,23,78,-23,78,-21,77
3480 DATA 25,31,121,122,127,153,255,8

```


ABOUT READ STATEMENTS

To retrieve the information stored in a DATA statement, the BASIC command READ is used.

It is followed by one or more variables separated by commas. READ takes the first piece of information from the first DATA statement and puts it into the variable that follows the READ statement. If there is a second variable, the next piece of information is taken from the same DATA statement and put into the second variable. This continues until all the variables following the READ statement are filled. If everything from the first DATA statement is read, the computer automatically jumps to the next one.

There must be as many pieces of information in your DATA statements as are to be READ. The variable type (string or number) must match the type of information being READ. Here is an example using the DATA statements from page 17:

120 READ N This line reads the number 8 from the DATA statement in line 100 and puts it in the variable N.

130 READ A,B,C,D\$,E\$,F\$,G\$

This line reads all the information in line 100 and some from 105. Notice how the string variable reads string information and the number variables read numbers.

140 READ H\$,I,J,K,L\$,M\$,O

This line reads all the remaining information in lines 105 and 110.

If information in a DATA statement does not match the type being READ, you will get a "TYPE MISMATCH" error.

When two commas follow one another without any information between them, a READ statement will take this as either an empty string or the number zero. Here is an example:

150 DATA 10,20,,15,GREG,,6,7

160 READ G,H,I,J,K\$,L\$,M,N

This program gives the variables the following information:

G=10

H=20

I=0

J=15

K\$="GREG"

L\$="" This is an empty string.

M=6

N=7

When a DATA statement has been totally READ, another READ command will result in an "OUT OF DATA ERROR". To avoid this, always match up the number of pieces of information with the number of variables that will READ them.

The RESTORE command will allow you to use the DATA statements again.

RED ALERT

by Simon Parker and Peter Lear



Expansion Required: None



ou have penetrated the enemy planet's refueling station. You must pilot your spacecraft through the cavern that houses the fuel dump. The terrain is rough and changes radically. Be very careful; the survival of your planet depends on you. Use the @ key to move your ship up, and the ; key to move down; P moves you left and * moves you right. Bomb the fuel dumps with the shift key. You gain 200 points for every exploded fuel tank and lose 50 points for dropping a bomb and missing one. Oh, and do avoid the counter-attacking aliens. Shoot them with your laser gun (key A) and get 50 points for every "kill"; however, if you can bomb an alien ship you will gain 250 points! Good hunting!

VARIABLES USED:

P	- ship position
G	- top of cavern
J	- bottom of cavern
CC, DD	- cavern height multiplying factors
B, C	- random numbers
K, I, L, M, Z	- counters
L2, P2	- key pressed
S, Y, A, A\$	- temporary data handlers
N, N1	- bomb position
SC	- score


```

1 PRINT"[SC]      RED ALERT  "
2 PRINT"      BY SIMON PARKER":PRINT"      AND PETER LEAR"
3 PRINT"[CD]TAKE YOUR SHIP THROUGH THE CAVE.HIT TANKS AND
  JETS FOR POINTS.[CD]"
4 PRINT"      USE THE KEYS:"PRINT"      @ -UP":PRINT"      ; -
  DOWN":PRINT"      P -LEFT"
5 PRINT"      * -RIGHT":PRINT"      A -LASER":PRINT"      SHIFT
  -BOMB"
6 PRINT"[CD]      PLEASE WAIT"
7 P=7904:J=5:G=5:CC=.5:DD=.9
8 GOSUB1000:GOSUB5000
9 TI$="000000"
10 PRINT"[SC][WH]"
20 POKE36879,8:POKE36878,15
30 B=RND(1)
31 IFTI$="000200"THENPOKE36879,104:CC=.65:DD=.8
32 IFTI$="000330"THENPOKE36879,57:CC=.75:DD=.7
35 C=RND(1)
36 IFB<CCTHENJ=J+1:GOTO40
37 J=J-1
40 IFC<CCTHENG=G+1:GOTO46
45 G=G-1
46 IFJ>0ANDG>10THENG=G-1:J=J-1
47 IFJ+G>17THENJ=J-1:G=G-1
48 IFJ<1THENJ=1
49 IFG<1THENG=1
50 FORK=1TOJ
60 POKE8185-22*K,20
61 NEXTK:IFB<.2ORB>.8THENPOKE8185-22*K,22:POKE8185+30720-22
  *K,2
62 FORK=1TOG
65 POKE7701+22*K,20
66 NEXTK
70 IFC>DDTHENPOKE7701+22*K+44,24:POKE38421+22*K+44,2
91 POKEP,32
100 SYS(829)

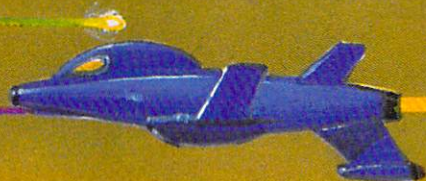
```




```

125 POKEP,21:POKEP-1,32
126 PRINT"[HM][RV]SCORE[RO]"SC
130 POKEP,32
135 L2=PEEK(197):P2=PEEK(653)
143 IFL=30ANDP1=119ORP2=1ANDL2=14THENGOSUB4000:GOTO181
145 IFL=30ORP2=1THENSC=SC-50:GOSUB3000:GOTO181
148 IFL2=17THENGOSUB4500
150 IFL=54ORL2=22THENP=P+22
160 IFL=46ORL2=13THENP=P-1
170 IFL=58ORL2=53THENP=P-22
180 IFP1=119ORL2=14THENP=P+1
181 IFPEEK(P+1)<>32THENGOTO2000
190 POKEP,21:POKEP-1,32:POKE36877,120+INT((P-7680)/4)
200 GOTO30
1000 POKE52,28:POKE56,28
1010 FORI=7168TO7679:POKEI,PEEK(I+25600):NEXT
1030 FORC=7328TO7367:READA:POKEC,A:NEXT
1035 POKE36869,255:RETURN
1040 DATA170,85,170,85,170,85,170,85
1050 DATA0,128,192,224,254,255,255,0
1060 DATA60,60,126,126,255,60,66,129
1070 DATA160,80,168,75,168,80,160,0
1080 DATA0,1,3,7,127,255,255,0
2000 FORI=1TO10
2010 POKE36879,I*10
2011 POKE36878,10-I
2012 POKE36877,220
2020 FORK=1TO200:NEXTK
2030 NEXTI
2040 FORL=1TO1000:NEXTL
2045 POKE36869,240:PRINT"[SC]"
2050 PRINT"[HM][CD][CD][RV]GAME OVER[RO]"
2060 PRINT"[CD][CD]YOU SCORED"SC
2070 PRINT"[CD][CD]ANOTHER GAME (Y/N)"
2080 GETA$:IFA$=""THEN2080

```




```

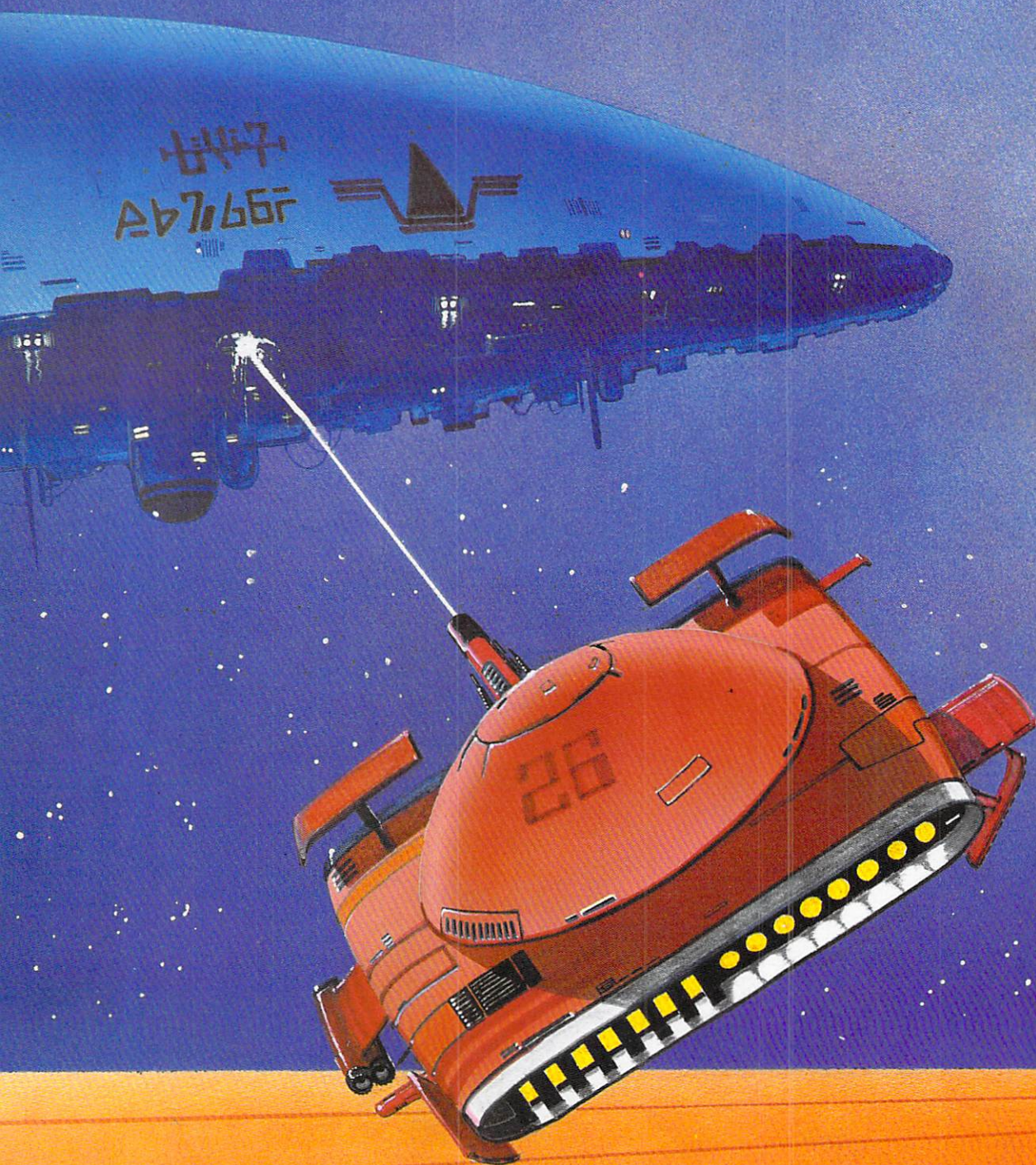
2081 IFA$="Y"THENJ=8:G=8:P=7904:SC=0:TIS="000000":POKE36869
,255:GOTO10
2082 IFA$="N"THENPRINT"[SC]":POKE36869,240:END
2090 GOTO2080
3000 FORM=1TO10
3005 POKEN,32
3010 N=P+M*22:POKE36876,150+M*10
3020 IFPEEK(N)=22THENSC=SC+250:POKEN,42:POKE36876,0:RETURN
3030 IFPEEK(N)=20THENPOKEN,42:POKE36876,0:RETURN
3040 POKEN,90:NEXTM:POKEN,32:POKE36876,0:RETURN
4000 SC=SC-100:FORB=1TO10:POKEM1,32:M1=P+B:POKE36876,240:
POKE36876,0
4010 IFPEEK(M1)=20THENPOKEM1,20:RETURN
4011 IFPEEK(M1)=22THENSC=SC+200:POKEM1,32:RETURN
4012 IFPEEK(M1)=24THENSC=SC+150:POKEM1,32:RETURN
4020 POKEM1,42:POKEM1+30720,1
4030 NEXTB:POKEM1,32:RETURN
4500 POKE36876,129
4505 FORZ=P+2TOP+7
4510 Y=PEEK(Z)
4515 POKEZ,46
4520 IFY=24THENSC=SC+50:Y=32
4525 IFY=22THENSC=SC+200:Y=32
4530 POKEZ,Y
4550 NEXTZ
4560 POKE36876,0
4590 RETURN
5000 FORK=829TO874:READS:POKEK,S:NEXTK:RETURN
5010 DATA169,19,32,210,255,169,29,32,210,255,169,13,32,210,
255,169,0,141,60,3
5020 DATA169,29,32,210,255,169,20,32,210,255,169,13,32,210,
255,238,60,3,173,60,3
5030 DATA201,21,208,231,96
5040 RETURN

```




INVASION

by Alex Kiernan and Peter Lear



Expansion required: None



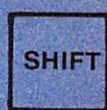
he future of your planet is in your hands. Yours is the last alpha base left, everything else has been destroyed by the Astrol Fleet. You can still defeat the warring aliens, but it will take skill and cunning on your part. Only one multi-directional laser is still operational, but due to a computer malfunction, you cannot move the laser and fire it at the same time. Z moves your base left, X moves it right; , moves your turret left and . moves it right. Fire your laser with the space bar. Remember, existence as you know it depends on you. Good luck!

Graphic Symbols Used

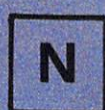
To Get:



Press Together:



&



&



&



VARIABLES USED:

N	- invader position
A\$ (0) - A\$ (2)	- player's ship
A (0) - A (2)	- turret character
Y	- player's position
P	- character pointer
R\$	- draws turret character
S\$	- moves cursor to bottom of screen
C(0) - C(2)	- turret angle
V	- difficulty factor
L, M, F,	
I, D, A	- counters
T, Q	- random numbers
A\$	- key pressed
X	- number of laser shots remaining
O	- laser position
Z	- character in laser's path


```

1 A$(0)="[RD]      "+CHR$(109)+"    [WH]":A$(1)="[RD]      "+CHR$(167
  )+"    [WH]":A$(2)="[RD]      "+CHR$(110)+"    [WH]"
2 A(0)=23:A(1)=22:A(2)=21:Y=5:P=0
3 POKE650,128:GOTO1000
5 R$="[CU][CL][CL][CL][CL] [RD] [WH] [CU][CL][CL][CL][CL]
  [CL][CL]":S$="[HM][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]
  [CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]"
6 C(0)=77:C(1)=103:C(2)=78:PRINT"[SC]"
9 FORA=1TO50-(V*X)/100:POKE7680+(RND(TI)*396),46:NEXT
10 FORD=0TO9
20 PRINTS$;"[GR][RV]                                [RO][HM][WH]"
40 N=RND(TI)*20+7760:GOTO170
50 POKEN,32:POKEN+1,32:POKEN+2,32
51 Q=INT(RND(TI)*V)
52 ON(Q>V)GOTO56
53 Q=INT(RND(TI)*2)*22:IFQ>0THENN=N+Q:GOTO56
55 N=N+INT(RND(TI)*2)*44-22:IFN<7680THEN40
56 FORI=1TOV
57 T=INT(RND(TI)*3)-1
58 IFT<>0THENN=N+T:I=V
59 NEXTI
60 POKEN,78:POKEN+1,86:POKEN+2,77
70 IFPEEK(N+22)<>32ANDPEEK(N+22)<>46THEN280
80 FORI=1TO200*V
90 GETA$:REM IFA$=""ANDT+V>TITHEN90
100 IFA$=""THEN50
110 IFA$="X"ANDY<18THENY=Y+1:P=0:GOTO170
120 IFA$="Z"ANDY>5THENY=Y-1:P=2:GOTO170

```




```

130 IFA$=","ANDP>0THENP=P-1:GOTO170
140 IFA$="."ANDP<2THENP=P+1:GOTO170
150 IFA$<>" "THEN50
160 GOTO180
170 PRINTSS;SPC(Y);R$+A$(P)
172 A$=""
177 GOTO50
180 O=8096+Y+P
185 X=X-1:IFX<0THEN170
190 POKE36878,10:FORF=8096+Y+PTO7720STEP-A(P):POKEF,C(P):
    POKEO,32:O=F
195 POKE36876,(F-7680)/2
200 Z=PEEK(F-A(P))
205 IFZ=46THEN220
207 IFZ<>32ANDZ<>46THENPOKEO,32:POKE36878,0:GOTO240
210 IF(F-7680)/22<>INT((F-7680)/22)THENNEXT
220 POKEO,32
225 POKE36878,0
230 GOTO170
240 POKEN,32:POKEN+1,32:POKEN+2,32:POKE36877,220:FORL=15TO0
    STEP-1:POKE36878,L
250 FORM=1TO100:NEXT
260 NEXT:POKE36877,0:POKE36878,0
270 NEXTD
275 GOTO2000
280 POKE36878,15:FORL=30TO35:FORM=220-LTO160-LSTEP-4:
    POKE36876,M:NEXTM
300 FORM=160-LTO220-LSTEP4:POKE36876,M:NEXTM,L:POKE36878,0:
    POKE36876,0

```




```

320 POKE36877,220:FORL=15TO0STEP-1:POKE36878,L:FORM=1TO100:
    NEXT:NEXT:POKE36877,0
330 POKE36878,0
340 GOTO1500
1000 POKE36879,8
1010 PRINT"[SC]          [YL]!INVASION!"
1020 PRINT"          =====[WH]"
1030 PRINT"[CD][CD][CD][CD][CD][CD][CD][CR][CR][CR][RD]BY A.
    G.KIERNAN[CD][CD][CD]"
1040 PRINT"          PRESS 'Y' FOR[CD]          INSTRUCTIONS OR ANY"
1050 PRINT"[CD]          KEY TO PLAY[WH]"
1060 GETA$:IFA$=""THEN1060
1070 IFA$<"Y"THEN1380
1100 PRINT"[SC][CD][CD] [YL]YOU ARE IN  CONTROL OF THE SOLE
    REMAINING ALPHA BASE ON THE"
1110 PRINT"TERRAN PLANET '[BL]EARTH[YL]'."
1120 PRINT"[CD]  IT IS YOUR JOB TO  PREVENT THE MARAUDING
    ASTROL BATTLE FLEET"
1130 PRINT"FROM INVADING THE      PLANET."
1140 PRINT"[CD] BUT DUE TO PREVIOUS  ATTACKS THE COMPUTER
    HAS MALFUNCTIONED."
1150 PRINT"[CD][WH]  PRESS ANY KEY"
1160 GETA$:IFA$=""THEN1160
1190 PRINT"[SC][CD][CD] [YL]DUE TO THIS MAL-      FUNCTION
    THE BASE  AND GUN TURRET MAY NOT";
1200 PRINT"BE MOVED TOGETHER"
1220 PRINT"[CD]  WHILE MOVING THE  COMPUTER RELEASES";
1230 PRINT"          CONTROL OF THE GUN  TURRET"
1240 PRINT"[CD][WH]  PRESS ANY KEY"
1250 GETA$:IFA$=""THEN1250
1280 PRINT"[SC][CD][CD] [YL]THE CONTROLS ARE:[CD][CD][WH]"
1290 PRINT"  Z-MOVES BASE LEFT"
1300 PRINT"  X-MOVES BASE RIGHT"

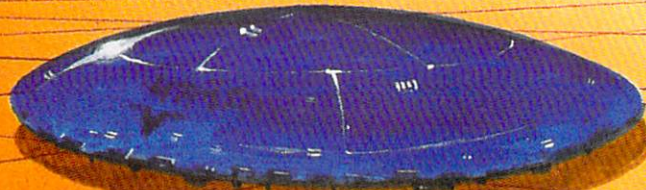
```




```

1310 PRINT" ,--MOVES TURRET LEFT"
1320 PRINT" .--MOVES TURRET RIGHT"
1330 PRINT" SPACE BAR FIRES"
1340 PRINT"[CD][YL] PRESS ANY KEY"
1350 GETA$:IFA$=""THEN1350
1380 PRINT"[SC][CD][CD] [YL]WHAT DIFFICULTY[CD]"
1385 PRINT"1:DIFFICULT - 9:EASY[CD]"
1390 INPUT" LEVEL [CL][CL][CL][CL]";A$:IFA$=""ORVAL
(A$)>9ORVAL(A$)<1THENPRINT"[CU]";:GOTO1390
1395 V=VAL(A$)*10
1400 PRINT"[CD][CD] WHAT IS YOUR[CD]"
1410 PRINT"1:EXCELLENT - 5:POOR"
1415 PRINT"[CD]MARKSMANSHIP RATING"
1420 INPUT" [CL][CL][CL][CL][CL]";A$:IFA$=""ORVAL
(A$)>5ORVAL(A$)<1THENPRINT"[CU]";:GOTO1420
1425 X=VAL(A$)*10
1430 PRINT"[WH][CD][CD]PRESS ANY KEY TO PLAY"
1440 GETA$:IFA$=""THEN1440
1450 GOTO5
1500 PRINT"[SC]!YOU'VE BEEN BLOWN UP!"
1510 IFD>0THENPRINT"[CD]BUT YOU SHOT DOWN"D SHIPS FIRST":
GOTO1530
1520 PRINT"AND YOU DIDN'T HIT A THING"
1530 PRINTSPC(5)" PRESS 'Y'[CD] TO PLAY AGAIN OR ANY"
1540 PRINT"[CD] KEY TO END[WH]"
1545 POKE198,0
1550 GETA$:IFA$=""THEN1550
1560 IFA$="Y"THEN1380
1570 END
2000 PRINT"[SC][RD]CONGRATULATIONS YOU SURVIVED"
2010 IFI=10THENPRINT"AND YOU SHOT THEM ALL":GOTO1530
2020 GOTO1510

```



HOW A GAME IS MADE

Every game starts in the same place... in someone's head. The idea is then put down on paper. All the features the game will have are written down. Pictures of the various characters and backgrounds are drawn. Every rule and aspect of the game is included in this paper plan.

The next step is to put all this information into an order of events. On another piece of paper shapes are drawn and each event of the game is put in a box, circle or diamond. Then, with each figure a brief note of purpose is made. The name of this series of shapes and notes is a flowchart.

Every event in the flowchart is a small program in itself. These small programs are commonly called subroutines. Breaking all the events into subroutines makes the task of programming the game much easier. Tracing a flaw in a particular subroutine is easier than tracking one down in a long, unco-ordinated program.

Quite often subroutines can be used more than once. They can even be transferred from one game to another. There is no point in designing a new subroutine to examine which way a player has moved the joystick for every game using a joystick. By using some of the same subroutines from game to game, a programmer will save him or herself a lot of time.

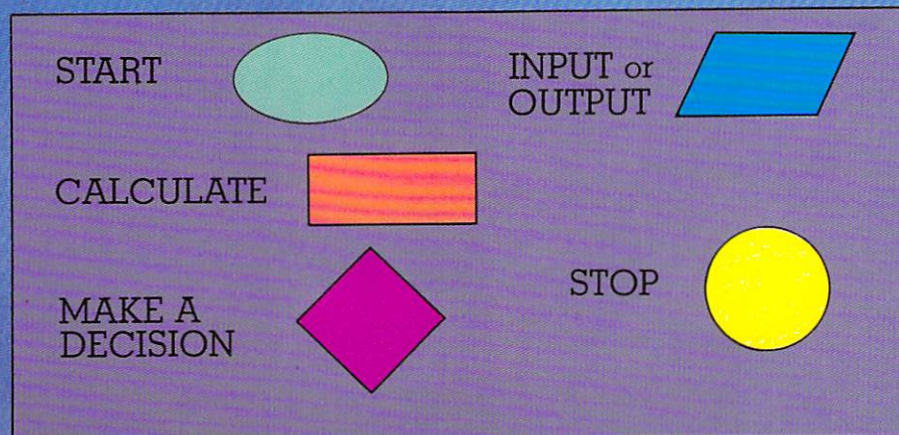
Here are some guidelines to follow when designing a game:

1. Write down your ideas.	2. Draw up a flowchart.
3. Use subroutines.	4. Use the same subroutines where possible.

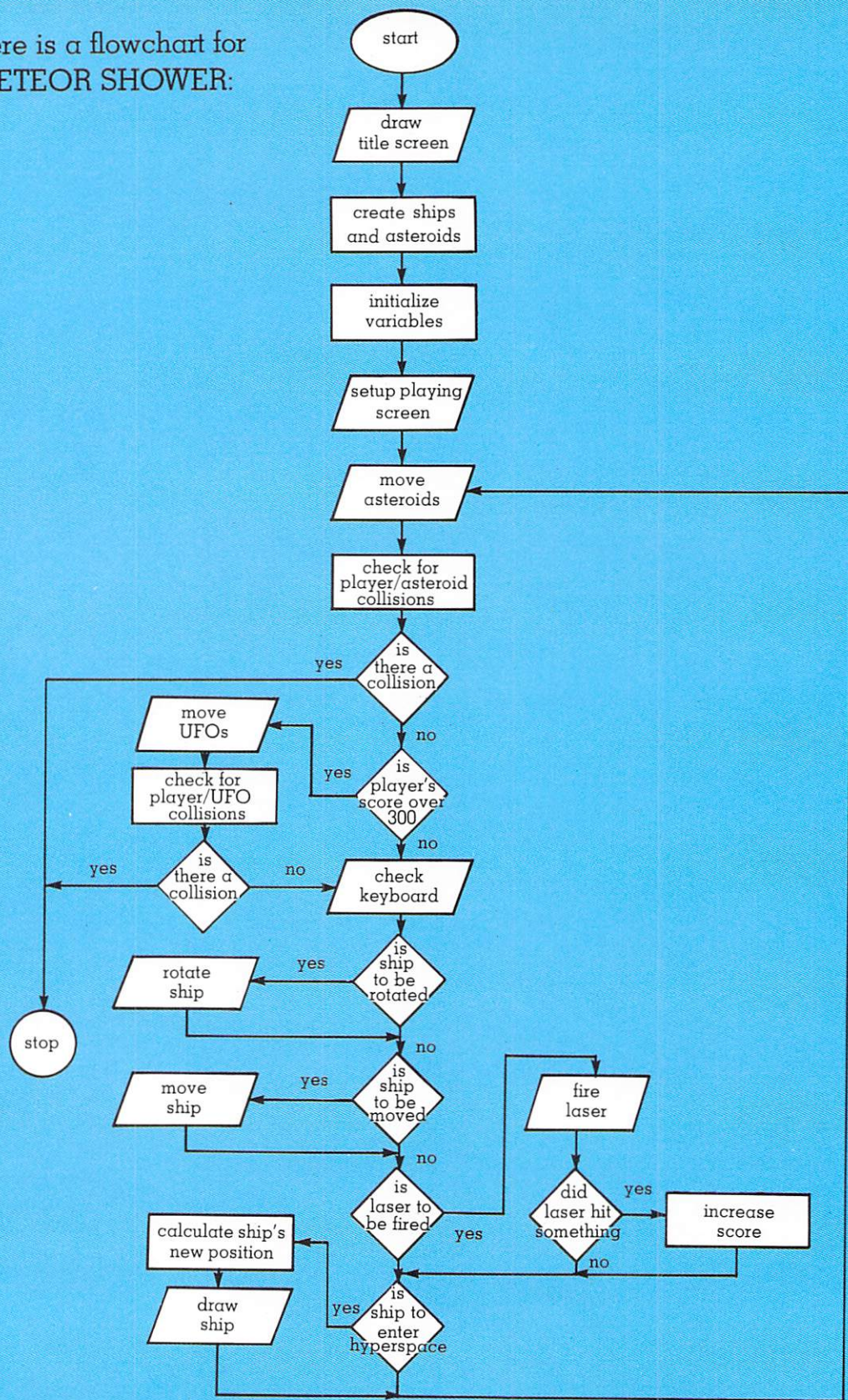
The game METEOR SHOWER started with these ideas:

- the player will be able to move the ship in a bounded area on the screen
- the ship will rotate on its axis
- the ship will be able to move forward to fire its laser in the direction in which it is pointing
- to challenge the player, asteroids will randomly appear and move across the screen
- points will be awarded for the laser destruction of asteroids
- the player may have the ship enter hyperspace (it will disappear and reappear elsewhere on the screen), but he will lose points for this
- when the player has reached a certain score, alien ships (UFOs) will also appear on the screen to menace him

These ideas were then summarized in a flowchart. The shapes in the flowchart mean:



Here is a flowchart for
METEOR SHOWER:



Here is a verbal description of how the game METEOR SHOWER works:

Line numbers	Operation
100 - 200	data statements store values for sound making and character set redesigning
210	sets top of BASIC memory, sets volume to maximum
220 - 250	changes screen color and prints title screen
260 - 270	plays sound
280	prints game introduction on screen
290	waits for a key to be pressed
300	creates spaceship and meteor shapes
310	moves some character sets from ROM to RAM
320	turns on the special characters
330 - 360	sets score to zero and puts ship in starting position, sets display colors
370 - 380	sets up screen boundaries
390 - 700	main game loop
400 - 500	checks asteroid positions for possible collisions
510	if player's score is over 300, starts putting UFOs on screen
520	checks UFO positions
530	draws another UFO
540	draws asteroids
550	makes UFO sound
560	ends game if an asteroid has hit the ship
570	turns sound off
580	checks the keyboard if asteroid path is not empty
590	draws asteroids
600 - 680	checks keyboard
670	prints score
690	draws ship
700	end of game loop
710 - 820	subroutine to move ship
830 - 870	fires laser
880	increases score if asteroid hit
890	prints score on screen
900	clears laser shot from screen
910	turns off laser shot sound
920	restarts game loop
930	ends game

By examining the ideas, the flowchart, the listing description and the program listing itself, you will be able to see how METEOR SHOWER fits together. You may follow these same steps when creating games of your own. **HAVE FUN!**

More
HAYES
GETTING AHEAD

Computer Books:

FANTASTIC GAMES

for the VIC 20

Six fantastic games to key into your own
VIC 20 computer!

FANTASTIC GAMES

for the Commodore 64

Six arcade style games. Challenging fun
for the novice and expert alike!

APPLE GRAPHICS

Learn to program your own graphics
games on the APPLE computer starting
with the great space game, Appleflight!

EXPLORING THE VIC

From hook-up to peripherals,
programming to expansion memory – the
VIC 20 explained! Includes a terrific
game!

HAYES PUBLISHING LTD.

3312 MAINWAY

BURLINGTON, ONTARIO

L7M 1A7

ISBN 0-88625-058-7

J.P. PBM Products By Mail
Box 60515, Jane/Wilson Postal Outlet
1597 Wilson Ave.
Downsview, Ontario M3L 1B0